Zeitschrift: Schweizerische Zeitschrift für Soziologie = Revue suisse de sociologie = Swiss journal of sociology

Herausgeber: Schweizerische Gesellschaft für Soziologie

Band: 23 (1997)

Heft: 3

Artikel: Technologische Paradigmen und lokaler Kontext : das Beispiel der ERMETH

Autor: Furger, Franco / Heintz, Bettina

DOI: https://doi.org/10.5169/seals-814630

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 21.08.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Schweiz. Z. Soziol./Rev. suisse sociol./Swiss Journ. Sociol., 23 (3), 1997, 533–566

TECHNOLOGISCHE PARADIGMEN UND LOKALER KONTEXT. DAS BEISPIEL DER ERMETH[1]

Franco Furger und Bettina Heintz
Institute of Public Policy, George Mason University, Fairfax, und
Institut für Soziologie, Universität Mainz

1. Einleitung

Der Digitalcomputer ist ein typisches Beispiel einer Mehrfacherfindung. Zwischen 1943 und 1946 wurden unabhängig voneinander vier verschiedene Prototypen entwickelt: der *Z3* von Konrad Zuse (1941), der *Mark I* von Howard Aiken (1944), das *Model V* von George Stibitz (1946) und schliesslich der erste elektronische Computer, der *ENIAC*, von Presper J. Eckert und John W. Mauchly (1946). Alle Computertypen, die zu dieser Zeit entwickelt wurden, hatten entscheidende Merkmale gemeinsam[2], in ihrer praktischen Ausführung unterschieden sie sich jedoch beträchtlich: der Mark I, das Model V und Zuses Z3 waren elektromechanische Maschinen, der ENIAC war ein elektronischer, auf Röhren basierender Computer. Der Mark I und der ENIAC (in seiner ersten Version) verwendeten das dezimale Zahlensystem, der Z3 und das Model V waren duale Maschinen. Das Rechenwerk von Mark I verarbeitete die Ziffern eines Wortes hintereinander, d. h. seriell, im ENIAC, im Z3 und im Model V wurden die Ziffern gleichzeitig, d. h. parallel verarbeitet. Der ENIAC (in seiner ersten Version) war aus heutiger Sicht ein Parallelcomputer, d. h. er konnte verschiedene Operationen gleichzeitig ausführen, im Z3, dem Mark I und dem Model V wurden die einzelnen Befehle nacheinander, d. h. sequentiell verarbeitet (Burks und Burks, 1981; Rutishauser, Speiser und Stiefel, 1951).

Praktisch alle Kombinationen kamen vor, und es war zu dieser Zeit noch keineswegs entschieden, welches Design das beste war. Anfang der 50er Jahre war der Selektionsprozess dann abgeschlossen. „Computer" hiess nun dual

1 Wir möchten den anonymen Reviewern für ihre freundlichen Kommentare danken.

2 Diese gemeinsamen Merkmale sind gleichzeitig die wichtigsten Definitionsmerkmale eines Digitalcomputers: (1) Alle vier genannten Maschinen operieren diskret, d.h. es sind *digitale* Maschinen. (2) Sie gehen bei der Durchführung einer Aufgabe *automatisch* vor, d. h. ohne menschliche Intervention. (3) Sie folgen in ihrem Vorgehen einer Folge von Befehlen, d. h. einem *Programm*. (4) Sie sind hinsichtlich ihres Funktionsbereichs *universell*, d. h. nicht auf eine bestimmte Aufgabe beschränkt. Es ist das Programm, das aus der universellen Maschine eine funktional spezifizierte macht.

(und nicht dezimal); elektronisch (und nicht elektromechanisch); zusammen mit den Daten gespeichertes Programm (und nicht separate Speicherung); parallele Datenverarbeitung (d. h. gleichzeitige Verarbeitung aller Ziffern eines Wortes) und nicht serielle; sequentielle Befehlsverarbeitung (und nicht parallele). Damit hatte sich eine Variante durchgesetzt, die sog. *Von-Neumann-Architektur*, die bis in die 70er Jahre die Computerentwicklung massgeblich prägte. Dies ist jedenfalls eine verbreitete Auffassung in der Computergeschichte.[3]

Gemäss dieser Sicht fand die entscheidende Entwicklung Mitte der 40er Jahre statt, als der bekannte Mathematiker John von Neumann in zwei Berichten die Grundprinzipien des modernen Digitalcomputers festlegte. Beim ersten Bericht handelt es sich um den 1945 geschriebenen *First Draft of a Report on the EDVAC* (von Neumann, 1945), beim zweiten um den zusammen mit Arthur Burks und Herman Goldstine verfassten Report *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, in dem das Grunddesign des am *Institute for Advanced Study* entwickelten IAS-Computers beschrieben wurde (Burks, Goldstine und von Neumann, 1946). Obschon die beiden Berichte erst Jahre später publiziert wurden, fanden sie informell rasche Verbreitung. Wer sich zu dieser Zeit mit Computern beschäftigte, und das war damals noch eine überschaubare Gruppe, deren Mitglieder sich praktisch alle persönlich kannten, hat die Berichte gelesen und sich an ihnen orientiert (zur Bedeutung von Neumanns für die Computerentwicklung vgl. Aspray, 1990a).[4] Zur Diffusion dieser Ideen haben nicht zuletzt auch die Konferenzen beitragen, die man seit dem Ende des 2. Weltkrieges zu organisieren begann, angefangen mit der berühmten Sommeruniversität, die 1946 an der *Moore School of Electrical Engineering*, wo der ENIAC gebaut worden war, durchgeführt wurde, bis hin zur Konferenz an der Universität in Manchester 1951 (Aspray, 1986)

Schon sehr bald hatte sich die von John von Neumann vorgeschlagene Konzeption durchgesetzt und wurde zu einem technischen und theoretischen

3 Zur Frühgeschichte des Computers gibt es eine umfangreiche, oft allerdings vorwiegend technisch orientierte Literatur, vgl. u. a. Aspray (1990a, 1990b); Ceruzzi (1983); Metropolis, Howlett und Rota (1980); Petzold (1985); Ramunni (1989); Randell (1982); Stern (1981); und zum militärischen Kontext der Computerentwicklung vgl. Cohen (1988).

4 Maurice Wilkes, der 1949 mit seinem EDSAC den ersten Computer mit einer Von-Neumann-Architektur entwickelt hatte, beschreibt in seiner Autobiographie, welchen Eindruck der *First Draft* damals auf ihn machte: „In it, clearly laid out, were the principles on which the development of the modern digital computer was to be based: the stored program with the same store for numbers and instructions, the serial execution of instructions, and the use of binary switching circuits for computation and control. I recognized this at once as the real thing, and from that time on never had any doubt as to the way computer development would go" (Wilkes, 1985, 108 f.).

Standard, der von nun an die Computerentwicklung bestimmte: „Les textes écrits autour de l'IAS [...] ont fondé la methodologie des ordinateurs" (Ramunni, 1989, 69). Oder wie es Paul Ceruzzi formuliert: „With some modifications the IAS design has survived as the paradigm of modern computer architectures" (Ceruzzi, 1983, 140).

2. Techniksoziologische Perspektiven

Die in der Computergeschichte verbreitete These, dass sich Anfang der 50er Jahre die Von-Neumann-Architektur als handlungsleitendes Paradigma durchgesetzt hat, ist mit der Annahme verknüpft, dass ein erfolgreiches technologisches Paradigma die weitere Entwicklung in hohem Masse determiniert. Wir möchten im folgenden am Beispiel eines in den frühen 50er Jahren in der Schweiz gebauten Computers exemplarisch aufzeigen, dass diese Annahme mit gewissen Einschränkungen zu lesen ist.

(1) *Determinismus vs. Kontingenz.* Während die Technikforschung der 50er und 60er Jahre Technikentwicklung als einen hochgradig determinierten Prozess betrachtet hatte, wird in der neueren Literatur zur Technikgenese die *Kontingenz* technischer Entwicklungsprozesse betont (vgl. Rammert 1995). Technische Entwicklung stellt sich nicht mehr als gradliniger Fort-Schritt dar hin zur immer perfekteren technischen Lösung, sondern als ein selektiver und diskontinuierlicher Prozess. Was sich in der Retrospektive als Erfundenes präsentiert, bezeichnet meistens *eine* Variante aus einem Bündel oft äquivalenter Lösungen. Jene Variante, die sich durchsetzt, begründet ein *technologisches Paradigma*: „In broad anology with the Kuhnian definition of a ‚scientific paradigm', we shall define a ‚technological paradigm' as a ‚model' and a ‚pattern' of solution of *selected* technological problems, based on *selected* principles derived from natural sciences and on *selected* material technologies" (Dosi, 1982, 152). Hat sich ein Paradigma einmal durchgesetzt, so legt es eine Bahn fest – eine „trajectory" –, die von nun an das Spektrum technischen Handelns eng umgrenzt: „A technological trajectory is the activity of technological progress along the economic and technological trade-offs defined by a paradigm" (Dosi, 1988, 225).[5]

Der in der Technikgeneseforschung prominent gewordene „sozialkonstruktivistische" Ansatz verfolgt eine ähnliche Argumentation (Pinch und Bijker, 1987). Obschon die Terminologie eine andere ist – so spricht Bijker (1987)

5 Zur Verwendung des Paradigmabegriffs in der Technikforschung vgl. auch van den Belt und Rip (1987); Constant (1973); Gutting (1984).

z. B. von „technologischen Rahmen" anstatt von Paradigmen[6] –, weicht der sozialkonstruktivistische Ansatz nicht grundsätzlich von Kuhns Modell ab. Ähnlich wie Kuhn unterscheiden auch Pinch und Bijker (1987) zwischen einer offenen Phase, in der verschiedene Alternativen miteinander konkurrieren – der Schlüsselbegriff ist der von Collins (1981) entlehnte Begriff der „interpretativen Flexibilität" –, und einer Phase der Stabilität, die einsetzt, nachdem sich ein Paradigma bzw. ein „technologischer Rahmen" durchgesetzt hat. „Closure in technology involves the stabilization of an artifact and the ‚disappearance' of problems. To close a technological ‚controversy', one need not to solve the problems in the common sense of that word. The key point is whether the relevant groups see the problem as being solved" (Pinch und Bijker, 1987, 44).

Die Schliessung einer technologischen Kontroverse kann verschiedene Ursachen haben. Sie kann Folge einer „amalgamation of vested interests" (Bijker, 1992, 94) oder auch das Resultat einer „translation" sein – eines komplexen Prozesses der Kooptation, Umdefinition und Allianzenbildung (Latour, 1987, Kap. 3; Callon, 1986). Während Pinch und Bijker den Aushandlungsprozess eher in Termini eines Interessenausgleichs beschreiben, sieht die Akteur-Netzwerk-Theorie, wie sie von Michel Callon, Bruno Latour und John Law vertreten wird, die Schliessung einer Kontroverse als Ergebnis eines subtilen Machtspiels, in dem nicht nur die beteiligten Menschen, sondern auch die nichtmenschlichen Objekte – die Mikroben (Latour 1988a), Muscheln (Callon 1986) und Katalysatoren (Callon 1983) – zu „Aktanten" werden. Auch wenn man das etwas überzogene Symmetriekonzept von Callon und Latour nicht teilt, so verweist der Netzwerkbegriff doch auf einen wichtigen Sachverhalt, der die Technikentwicklung von der Wissenschaftsentwicklung unterscheidet. Während die Träger wissenschaftlicher Wissensproduktion sozial relativ homogen sind, findet Technikentwicklung in der Regel in einem heterogenen sozialen Setting statt. In Technikprojekten treffen Akteure aus verschiedenen sozialen Bereichen zusammen, die unterschiedliche institutionelle Standpunkte vertreten – ForscherInnen aus dem universitären Bereich, Techniker aus dem privatwirtschaftlichen Forschungs- und Entwicklungs-Bereich, Produzenten und Manager, KonsumentInnen und unter Umständen auch staatliche Instanzen. Der Erfolg eines Technikprojekts hängt folglich entscheidend davon ab, inwieweit es gelingt, eine gemeinsame Sprache zu finden und zwischen den verschiedenen Interessen zu vermitteln.

6 Der Hauptunterschied besteht darin, dass Bijker auch nicht-professionelle Gruppen, z. B. Konsumenten, in seine Analyse einbezieht, während bei Kuhn ausschliesslich WissenschaftlerInnen die Träger eines Paradigmas sind.

Die Techniksoziologie hat verschiedene Faktoren benannt, die eine solche Integrationsfunktion erfüllen können – angefangen bei „Leitbildern" (vgl. u. a. Dierkes, Hoffmann und Marz 1992; Dierkes u. a. 1995), denen gerade in heterogenen Technikprojekten eine ähnlich integrierende Funktion zukommt wie den „boundary objects" in der Wissenschaftssoziologie des Symbolischen Interaktionismus (u. a. Star und Griesemer 1991), bis hin zur Etablierung von Innovationsnetzwerken, die von Kowol und Krohn (1995) als eigenständiger Integrationsmodus neben Markt und formaler Organisation behandelt werden.

Das sozialkonstruktivistische Modell, wie es von Pinch und Bijker formuliert worden ist, beruht letztlich auf der Annahme, dass mit der Durchsetzung eines technologischen Modells die weitere technische Entwicklung weitgehend festgelegt ist. Obschon Bijker (1992) konzediert, dass Stabilisierung ein gradueller Prozess und unter bestimmten Bedingungen auch umkehrbar ist, ist die Vorstellung doch vorherrschend, dass sich nach der Durchsetzung eines „technologischen Rahmens" die Handlungsalternativen drastisch verringern. Dies schlägt sich auch in der Empirie nieder: die meisten Untersuchungen konzentrieren sich auf die offene Phase und den Prozess der Schliessung, während die nachfolgenden Entwicklungen vergleichsweise unterbelichtet bleiben.[7] So gesehen zeichnet sich die sozialkonstruktivistische Sicht auf die Technik durch ein eigentümliches Mischungsverhältnis von Konstruktivismus und Determinismus aus: eine starke Betonung von Kontingenz und Offenheit bis zum Moment der Schliessung, und impliziter Determinismus, sobald sich ein technologisches Paradigma durchgesetzt hat.

(2) *Universalität vs. Kontextualität.* Was in der Techniksoziologie bislang erst am Rande thematisiert wird – die Partikularität und Kontextabhängigkeit technischen Handelns – ist in der Wissenschaftssoziologie schon seit längerem ein Thema (vgl. als Überblick Heintz 1997). Karin Knorr-Cetina spricht in diesem Zusammenhang von der „kontextuellen Kontingenz" wissenschaftlichen Handelns (Knorr-Cetina, 1984, 33). Die Entscheidungen, die Wissenschaftler und Wissenschaftlerinnen treffen, sind in entscheidendem Masse von lokalen Opportunitäten abhängig und haben nicht den universellen Charakter, den ihnen die Wissenschaftstheorie zu- bzw. vorschreibt. Ob ein Experiment repliziert wird, ist nicht nur eine Frage des „scientific ethos", sondern auch abhängig davon, ob die dazu benötigten Apparaturen vorhanden sind. Die getroffenen Entscheidungen sind „idiosynkratisch", indem sie geprägt sind

7 Dies gilt nicht für Arbeiten, die sich stärker an der Innovationsforschung orientieren und aus diesem Grunde auch den Anwendungskontext in den Blick nehmen. Beispielhaft für eine solche Optik, die im Gegensatz zum sozialkonstruktivistischen Ansatz auch das Geschehen nach der „Schliessung" thematisiert, sind Asdonk, Bredeweg und Kowol (1991), die Erfindung als einen rekursiven Prozess zwischen Entwicklungs- und Anwendungskontext definieren. Ein gutes Beispiel für einen solchen rekursiven Prozess ist das up-grading von Programmen.

durch die lokale Kultur – durch die spezifischen Verfahrensweisen, Wertorientierungen und Prioritäten der jeweiligen Forschergruppe –, und sie sind „opportunistisch", indem sie sich nach den Möglichkeiten und Restriktionen des jeweiligen Kontextes richten (Knorr-Cetina, 1984, 63 ff.).

Damit stellt sich die Frage, wie es gelingt, solche lokalen Erkenntnisprodukte in universelle wissenschaftliche Tatsachen zu verwandeln, denen, wie es Karl Mannheim einprägsam formulierte, keine „Spuren menschlicher Herkunft" mehr anzusehen sind (Mannheim, 1931, 256). Die konstruktivistische Wissenschaftssoziologie hat ein beachtliches Spektrum an Universalisierungsstrategien zusammengetragen und gleichzeitig gezeigt, in welchem Ausmass diese Strategien sozial imprägniert sind: angefangen bei Bruno Latours „immutable mobiles" (Latour 1988b) und Joan Fujimuras „standardized packages" (Fujimura 1992) über den Einsatz von Rhetorik (u. a. Latour, Woolgar, 1979, 69 ff.; Knorr-Cetina, 1984, 175 ff.) und die Standardisierung von Masseinheiten und Messnormen (O'Connell 1993) bis hin zur bereits erwähnten Politik der „Translation". Während das Spannungsverhältnis zwischen lokaler Produktion und universellem Anspruch zu den Kernthemen der Wissenschaftssoziologie gehört, hat diese Frage in der Technikforschung noch nicht zu einer vergleichbaren Aufmerksamkeit geführt.[8]

(3) *Wissen vs. Handeln.* Neben der Betonung kontextueller Faktoren gibt es in der Wissenschaftssoziologie noch eine andere Entwicklung, die für die Techniksoziologie eine theoretische Leitlinie sein könnte: der Übergang von einem wissensanalytischen Zugang zu einem *pragmatischen* Handlungsmodell (vgl. Heintz 1993a). Die „neuere" Wissenschaftssoziologie, wie sie in den 70er Jahren in deutlicher Abgrenzung zur institutionalistisch orientierten Wissenschaftssoziologie Mertonscher Prägung entstanden ist, verstand sich zunächst einmal als *Wissens*soziologie – als „sociology of scientific *knowledge*". Im Anschluss an die Arbeiten von Thomas Kuhn ging es darum, das (natur-)wissenschaftliche Wissen selbst einer soziologischen Betrachtung zugänglich zu machen.[9] Wissenschaftliches Wissen wurde dabei gleichzeitig –

8 Eine Ausnahme ist Mikael Hard, der die Universalitätsdoktrin in Hinblick auf die Technik*entwicklung* einer deutlichen Kritik unterzog (Hard 1994) sowie Arbeiten, die sich mit der Soziologie des Tests befassen. Der Test eines neuen technischen Artefakts wirft an sich die gleichen Probleme auf wie ein Experiment. Jeder Test, z. B. eines neuen Flugzeugs, erfolgt unter spezifischen lokalen Voraussetzungen: auf einer bestimmten Teststrecke, unter spezifischen klimatischen Bedingungen, mit einer besonderen Crew etc. Wie kann man sicher sein, dass das Flugzeug auf einer anderen Teststrecke, unter anderen klimatischen Bedingungen und mit einer anderen Crew genauso fehlerfrei funktioniert? Und wie gelingt es den Technikentwicklern, bei der Präsentation ihrer Ergebnisse von der Kontextabhängigkeit ihrer Versuche zu abstrahieren? Zum Universalisierungsproblem im Bereich des Testens vgl. exemplarisch MacKenzie (1989).

9 Diese wissenssoziologische Wende in der Wissenschaftssoziologie traf sich mit einer parallelen Entwicklung in der Techniksoziologie. Während in den früheren, vorwiegend ökonomisch

und in konsequenter Weiterführung der von Kuhn eingeführten Gewichtung – vorwiegend als *theoretisches* Wissen definiert, die empirische Ebene wurde dagegen kaum thematisiert (vgl. zu dieser Gewichtung Galison 1988). Seit Mitte der 80er Jahren ist eine deutliche Abkehr von diesem wissenszentrierten Modell festzustellen. Die zentralen Referenzfiguren sind nicht mehr Thomas Kuhn oder Karl Mannheim, sondern die Klassiker des Pragmatismus, John Dewey, William James und Charles Sanders Peirce (und erstaunlicherweise nicht George Herbert Mead). Anstatt Wissenschaft ausschliesslich unter dem Aspekt des theoretisch formulierten Wissens zu betrachten, wird das praktische *Forschungshandeln* in den Vordergrund gerückt. Im Mittelpunkt steht nicht mehr das Wissen, sondern das Tun, nicht mehr die Theorie, sondern das Experiment – „from science as knowledge to science as practice", so die programmatische Überschrift zu einem Sammelband, der die pragmatische Wende in der Wissenschaftssoziologie breit dokumentiert (Pickering 1992). Die „pragmatische" Wende in der Wissenschaftssoziologie hat auf die Techniksoziologie noch kaum Auswirkungen gehabt. Die Arbeiten zur Technikgenese, und das gilt besonders für den sozialkonstruktivistischen Ansatz, sind nach wie vor weitgehend wissenssoziologisch orientiert. Dies ist umso erstaunlicher, als sich gerade im Bereich der Technikentwicklung eine Fokussierung auf die *Praktiken* ganz besonders aufdrängen würde.[10] Es gibt bislang erst eine einzige Studie, die sich ihrem Gegenstand nicht über Interviews oder die Analyse von Texten nähert, sondern auf teilnehmender Beobachtung der Tätigkeiten von Ingenieuren beruht, ähnlich wie es die Laborstudien für die Wissenschaft getan haben (vgl. Bucciarelli 1994).

Wir möchten im folgenden am Beispiel der Entwicklungsgeschichte eines in den 50er Jahren in der Schweiz gebauten Computers (ERMETH) diese Argumentation weiter ausführen. Am Beispiel der ERMETH lässt sich zeigen,

---

orientierten Arbeiten die Technik als „black box" behandelt worden war, wurde in den siebziger Jahren vermehrt die Forderung aufgestellt, Technik auch von *innen* her zu analysieren. Nathan Rosenbergs Buchtitel *Inside the Black Box* steht programmatisch für diesen Perspektivenwechsel in Technikgeschichte und Techniksoziologie (Rosenberg 1983). Technik wurde dabei primär kognitiv definiert: *Technology as Knowledge*, wie der Titel des nunmehr klassischen Aufsatzes von Edwin Layton hiess (Layton 1974), und technische Artefakte avancierten zu „geronnenen kognitiven Systemen" (Weingart, 1982, 114). Diese interne Umorientierung erklärt, weshalb der wissenszentrierte Ansatz der Wissenschaftssoziologie von der Techniksoziologie und -geschichte so schnell und breit rezipiert wurde. Vgl. exemplarisch zu dieser wissensorientierten Konzeption von Technik den von Rachel Laudan herausgegebenen Sammelband (Laudan 1984).

10 Mit „Praktiken" sind hier natürlich nicht bloss manuelle, sondern auch kognitive Praktiken gemeint – z. B. Berechnungen, das Schreiben von Programmen, das Anfertigen von Design- und Konstruktionsplänen etc. Die Analyse kognitiver Praktiken ist allerdings um einiges anspruchsvoller als die Rekonstruktion von „Handarbeit". Dies zeigt sich auch in der Wissenschaftssoziologie, wo es auch nach der „pragmatischen Wende" noch kaum Untersuchungen zur konzeptuellen Praxis gibt (vgl. Heintz 1998).

dass auch nach der Durchsetzung eines technologischen Paradigmas eine Reihe von Problemen offen bleiben, für die noch keine bereits erprobten Lösungsmuster bestehen. Ein technologisches Paradigma, wie z. B. die Von-Neumann-Architektur im Falle der Computerentwicklung, formuliert allgemeine Organisationsprinzipien, die nicht den Status von Konstruktionsalgorithmen haben, sondern einen breiten Handlungsspielraum offen lassen. Wie dieser Handlungsraum gefüllt wird, d. h. welche Entscheidungen konkret getroffen werden, hängt in starkem Ausmass auch von lokalen und situativen Faktoren ab.

Der Bau eines Computers erfordert Entscheidungen auf verschiedenen Ebenen – auf der technischen Ebene wie auf der Ebene der funktionalen Organisation.[11] Auf der *technischen* Ebene werden Entscheidungen getroffen, die sich z. B. auf das Material beziehen oder auf die Realisierung der elektrischen Schaltungen: nimmt man, um beim ERMETH-Beispiel zu bleiben, Quecksilber-Ultraschalleitungen, Magnettrommeln oder Williamsröhren für den Bau des Speichers (Speiser 1985)? Wie sollen die logischen Funktionen elektronisch implementiert werden (Schai 1957)?[12] Auf der Ebene der funktionalen Organisation geht es um die eigentliche *Architektur* des Computers. Entscheidungen auf dieser Ebene beziehen sich auf Fragen wie: will man ein duales oder ein dezimales Zahlensystem, Programmspeicherung oder externe Programmierung, Ein- oder Mehradressmaschinen, Gleitkomma oder Fixkomma? Welche Befehle, und wieviele?

Die verschiedenen Ebenen sind nicht völlig unabhängig voneinander. Eine Reihe von konzeptionellen Entscheidungen, die John von Neumann und seine Mitarbeiter getroffen haben, sind das Ergebnis technischer Restriktionen. Die Entscheidung beispielsweise, den EDVAC mit einem seriellen Rechenwerk auszurüsten, war wesentlich durch die damals verfügbare Speichertechnologie diktiert (mercury delay lines). Diese Abhängigkeit der konzeptionellen von der technischen Ebene verweist auf einen wichtigen Unterschied zwischen wissenschaftlichen und technologischen Paradigmen: während die Formulierung eines wissenschaftlichen Paradigmas empirischen und logischen Konsistenzkriterien zu genügen hat, ist der Spielraum bei der Entwicklung eines technologischen Paradigma eingeschränkt durch die verfügbare Technologie. Gesamthaft gesehen scheinen die Freiheitsgrade bei wissenschaftlichen Paradigmen um einiges zahlreicher zu sein als bei technologischen (Vincenti 1995).

11 Die Struktur eines Computers ist natürlich um vieles komplizierter, und insbesondere sind auch sehr viel mehr Ebenen – bzw. „Schichten" – zu unterscheiden. Vgl. zur Schichtenstruktur des Computers die Darstellung und die Diagramme in Giloi, 1981, 12 ff., und etwas weniger technisch Newell, 1986, 35 f.; Winograd und Flores, 1986, 86 ff.

12 Wie dies praktisch realisiert wird, beschreibt Tracy Kidder (1984) sehr anschaulich in seinem Report über den Bau eines Computers.

3. Das Projekt ERMETH[13]

1948 wurde an der Eidgenössischen Technischen Hochschule (ETH) in Zürich ein *Institut für Angewandte Mathematik* gegründet mit dem expliziten Ziel, die Einführung des „programmgesteuerten Rechnens" in der Schweiz zu fördern und für Mathematik, Naturwissenschaft und Technik nutzbar zu machen. Leiter dieses Institutes war der Mathematiker Eduard Stiefel, der sich ursprünglich als Topologe einen Namen gemacht hatte, während des 2. Weltkriegs mit Berechnungsproblemen in Kontakt gekommen war und schon sehr früh die Bedeutung des Computers für die Mathematik, speziell für die numerische und angewandte Mathematik erkannt hatte (vgl. Gutknecht, 1987, 63). Im gleichen Jahr wurde unter dem Vorsitz von Stiefel eine *Kommission zur Entwicklung von Rechengeräten in der Schweiz* gebildet, die, wie Stiefel in einem Gesuch an die *Eidgenössische Stiftung zur Förderung der schweizerischen Volkswirtschaft durch wissenschaftliche Forschung* festhielt, zur Auffassung gelangte, „dass die Einführung des wissenschaftlichen programmgesteuerten Rechnens in der Schweiz von grösster volkswirtschaftlicher Tragweite für die Entwicklung unserer Technik und Industrie" ist (Stiefel, 1949a, 3).[14] Aus diesem Grunde wurde schon sehr bald beschlossen, einen eigenen Computer zu bauen. Der Auftrag ging erstaunlicherweise nicht an eine technische Abteilung, sondern an das *Institut für angewandte Mathematik*, dessen Personal im Jahr 1949 aus zwei Mathematikern und einem Elektroingenieur bestand (Jahresbericht 1949/50).

Um sich die notwendigen Kompetenzen zu beschaffen, unternahm Stiefel 1949 mit seinen beiden Assistenten, dem Mathematiker Heinz Rutishauser und dem Elektroingenieur Ambros Speiser, eine ausgedehnte Studienreise durch

13 Die folgenden Ausführungen beruhen auf einer Reihe von Interviews, die wir im Winter und Frühjahr 1994 mit Personen geführt haben, die massgeblich an der Entwicklung der ERMETH beteiligt waren. Als weitere Informationsgrundlage beziehen wir uns auf zeitgenössische Publikationen sowie auf nicht veröffentlichte Quellen, die uns freundlicherweise von unseren Gesprächspartnern zur Verfügung gestellt wurden (Jahresberichte des *Instituts für angewandte Mathematik*, Protokolle, Gesuche, Memoranden, Briefe etc.). Wir möchten an dieser Stelle allen Gesprächspartnern für die Zeit und Mühe danken, die sie sich genommen haben.

14 Die Kommission setzte sich aus Vertretern der ETH und der Privatwirtschaft zusammen. Finanziert wurde die ERMETH über die oben erwähnte „Volkswirtschaftsstiftung" sowie durch private Firmen. Begründet wurde der Bau der ERMETH einerseits mit innerwissenschaftlichen Argumenten (Förderung der angewandten und numerischen Mathematik), andererseits in Hinblick auf ihren volkswirtschaftlichen und militärischen Nutzen. Das ERMETH-Projekt kann als ein relativ früher Versuch betrachtet werden, ein staatlich gefördertes Innovationsnetzwerk aufzubauen. Der Erfolg war allerdings bescheiden. Die ERMETH hat zwar einen wichtigen Beitrag geleistet für die Einführung des automatischen Rechnens in der Schweiz, es gelang jedoch nicht, sie kommerziell zu verwerten. Ein ähnliches, allerdings sehr viel grösseres Projekt, das zur selben Zeit initiiert wurde, war die staatliche Förderung der Atomtechnologie, vgl. zu diesem Projekt und seinem Scheitern Hug 1997.

die Vereinigten Staaten (Stiefel 1949b; Speiser, 1992, 5). Da es zu dieser Zeit kaum entsprechende Fachliteratur gab, war dies praktisch die einzige Möglichkeit, sich das Wissen zu beschaffen, das für den Bau eines Computers notwendig war. Wie William Aspray gezeigt hat, vollzog sich der Technologietransfer bis in die 50er Jahre, d. h. bis zur Kommerzialisierung des Computerbaus, vor allem über informelle Gespräche und persönliche Kontakte (Aspray 1986). Die Zentren der Computerentwicklung in den USA waren zu dieser Zeit die *Harvard University*, wo Aiken an seiner Mark-Serie arbeitete, und das *Institute for Advanced Study*, wo John von Neumann und seine Mitarbeiter die IAS-Maschine entwickelten – „die schnellste und mathematisch intessanteste elektronische Maschine", wie Stiefel in seinem Reisebericht schreibt (Stiefel, 1949b, 2). Rutishauser und Speiser hielten sich je ein halbes Jahr an diesen beiden Instituten auf.[15]

Da der Bau eines Computers mehrere Jahre in Anspruch nahm, wurde nach einer Übergangslösung gesucht. Nachdem zuerst die Miete einer IBM-Lochkartenmaschine (noch kein Computer) im Gespräch war, mietete die ETH 1950 den einzigen Computer, den es zu dieser Zeit auf dem Kontinent gab, den von Konrad Zuse 1945 fertiggestellten Z4 (vgl. u. a. Zuse, 1984, 104 ff.; Engeler u. a. 1981; Stiefel 1953).[16] Da man zunächst mit der Z4 Erfahrungen sammeln wollte, wurde erst 1952 mit der Entwicklung des eigenen Computers begonnen. Die ERMETH (Elektronische Rechenmaschine der Eidgenössischen Technischen Hochschule) wurde 1956 in Betrieb genommen und stand bis zur Anschaffung eines kommerziellen Computers im Jahre 1963 zur Verfügung. Ihre Rechenleistung war hundertmal grösser als die Leistung der Z4 (Memorandum o. J.).

Der Erfolg der ERMETH ist umso erstaunlicher, als an ihrer Entwicklung mehrheitlich junge, gerade erst diplomierte Mathematiker und Ingenieure beteiligt waren, die von ihrer Ausbildung her kaum über das mathematisch-technische Wissen verfügten, das für den Bau eines Computers notwendig

15 Nach Angaben von Stiefel waren 1949 in den USA 10 Computer fertiggestellt und 5 im Bau (Stiefel 1949b, 6). In der Übersicht von Rutishauser, Speiser und Stiefel werden für das gleiche Jahr 8 im Bau befindliche Computer erwähnt (Rutishauser, Speiser und Stiefel, 1951, 96 f.). Nicht erwähnt wird der zu dieser Zeit am MIT gebaute, aber unter Geheimhaltung stehende Computer *Whirlwind*, vgl. dazu Smith (1976). Vgl. auch die – allerdings etwas differierenden – Angaben zur Computerdichte in Aspray (1986) und Ceruzzi (1986).

16 Die Z4 wurde zwischen 1950 und 1955 für mehr als 50 verschiedene Problemstellungen eingesetzt, z. B. für Berechnungen im Bereich der Ballistik, der theoretischen Physik (quantenmechanische Untersuchungen an Naphtalinmolekülen), der Baustatik (Berechnung einer Staumauer) sowie für zahlreiche mathematische Untersuchungen, vgl. Schwartz, 1981, 123 ff., sowie die vom *Institut für Angewandte Mathematik* 1955 herausgegebene Zusammenstellung der mit der Z4 ausgeführten Aufträge.

war.[17] Lediglich Heinz Rutishauser verfügte über Erfahrung im Bereich der numerischen und angewandten Mathematik, und der für die Hardware zuständige Ambros Speiser war noch als Student 1948 nach Cambridge zu Maurice Wilkes gefahren, um dort etwas über den Bau von Computern zu lernen.[18] Die Boolesche Algebra, die heute zum festen Bestandteil der Informatik- und Elektrotechnik-Ausbildung gehört, war zu Beginn des Projektes niemandem bekannt, sondern musste von den Projektteilnehmern erst gelernt werden. Fast ebenso unbekannt war die Digitaltechnik. Die Elektrotechnik war damals noch primär auf Energietechnik und Telephonie ausgerichtet und vermittelte nur am Rande jenes Wissen, das man für den Bau eines Computers benötigte.

Auch die finanzielle Ausstattung war vergleichsweise bescheiden. Die ERMETH wurde von der ETH sowie durch Beiträge der Schweizerischen Volkswirtschaftsstiftung und privater Firmen bzw. Stiftungen finanziert. Über die in den verschiedenen Projekten verwendeten finanziellen Mittel stehen uns zwar keine gesicherten Zahlen zur Verfügung, die Anzahl der verwendeten Röhren kann jedoch als Indikator dienen: die ERMETH war die kleinste der bekannten Maschinen. Sie bestand aus circa 1800 Röhren. Die meisten anderen Computer waren um einiges grösser: Der IAS bestand aus 2300 Röhren, der von Maurice Wilkes gebaute EDSAC aus 3000, der EDVAC aus 3600 und der ENIAC aus 18000 Röhren.

Ein wesentlicher Faktor für den Erfolg des Projektes scheint die pragmatische Zielsetzung gewesen zu sein. Im Gegensatz zum IAS-Computer, mit dem man vor allem Erfahrungen im Bereich des Computerbaus sammeln wollte und der in diesem Sinne vorwiegend experimentellen Charakter hatte, stand bei der ERMETH nicht die Computerentwicklung selbst im Vordergrund. Der Bau der ERMETH hatte eine primär instrumentelle Funktion: es ging darum, das „programmgesteuerte Rechnen" und mit ihm die angewandte Mathematik zu fördern, und dazu brauchte man einen Computer. Referenzsystem war die Mathematik, nicht die Computertechnik.[19] Diese instrumentelle Haltung er-

17 1953 setzte sich das Team aus zwei Mathematikern (Stiefel, Rutishauser), vier Ingenieuren (Speiser, Schlaeppi, Stock und Appenzeller), einem Zeichner (Sieberling) und zwei Mechanikern (Müller, Messerli) zusammen; 1954 aus zwei Mathematikern, sechs Ingenieuren (zusätzlich Schai und Bielek), einem Zeichner, vier Mechanikern und einem Physiker, vgl. Jahresberichte 1953 und 1954. Mit Ausnahme eines Ingenieurs (Stock) hatten alle Beteiligten an der ETH studiert.

18 In der Schweiz war zu dieser Zeit über Computer nicht viel zu erfahren. In der NZZ erschien der erste Artikel 1948. Er war von Stiefel verfasst (Stiefel 1948). Die nächsten Artikel erschienen erst zwei Jahre später – ein Artikel von Rutishauser zu Aikens Computer (Rutishauser 1950) und von Speiser zur gerade in Betrieb genommenen Z4 (Speiser 1950).

19 Diese instrumentelle Haltung hat sich später nachteilig auf die Institutionalisierung der Informatik an der ETH ausgewirkt: Gebiete, bei denen die Maschine selbst im Mittelpunkt stand, wie etwa Programmiersprachen, Programmiertechnik, Künstliche Intelligenz etc., wurden bis Ende der 70er Jahre tendenziell vernachlässigt.

klärt, weshalb sich Stiefel in technischen Fragen stark an Aiken orientierte, obschon er die IAS-Maschine als die mathematisch interessantere betrachtete.[20] Die von Aiken entwickelten Maschinen zeichneten sich durch ein „konservatives" Design aus: maximiert wurde nicht die Schnelligkeit, sondern die Zuverlässigkeit. Entsprechend verwendete Aiken nicht die neuesten, sondern die erprobtesten Techniken und Ideen – mit der Folge allerdings, dass er mit der Zeit den Anschluss an die Computerentwicklung verpasste (zu Aiken vgl. Cohen 1987).

4. Handlungsspielräume und praktische Entscheidungen

Die ERMETH übernahm zwar gewisse architektonische Prinzipien des IAS-Computers, war aber nicht mit ihm identisch: (1) Sie bestand, wie es John von Neumann in seinem *First Draft* festgelegt hatte, aus fünf „Organen" – einem Leitwerk, einem Rechenwerk, einem Speicher und einem Eingangs- bzw. Ausgangsgerät.[21] (2) Die Befehle wurden hintereinander, d. h. sequentiell ausgeführt. (3) Die Befehle wurden zusammen mit den Daten gespeichert und nicht separat, wie es bei den Aiken-Maschinen der Fall war. Als Schreib- und Lesespeicher diente eine Magnettrommel. (4) Im Gegensatz zum IAS-Computer benutzte die ERMETH jedoch kein duales, sondern ein dezimales Zahlensystem. Und (5) arbeitete ihr Rechenwerk seriell, im Gegensatz zum IAS-Computer, der ein paralleles Rechenwerk hatte. Die ERMETH war offensichtlich kein Abbild der IAS-Maschine, sondern war auch durch andere Vorbilder geprägt und zeichnete sich zudem durch eine Reihe von Eigenentwicklungen aus.

20 Wir beziehen uns hier vor allem auf die Haltung Stiefels, der die Entscheidungen wesentlich bestimmte. Für die jungen Ingenieure im Team war die Zielsetzung in vielen Fällen eine andere. Sie waren mehr an der Computerentwicklung selbst interessiert, aber Stiefel war, wie es einer von ihnen formulierte, „in der Schweiz das einzige Vehikel", um an dieser Entwicklung teilnehmen zu können. Entsprechend dieser mehr technischen Zielsetzung befürworteten sie teilweise ein innovativeres Design, das Raum für die Erprobung neuer Ideen offen gelassen hätte. D. h. im Team gab es einen gewissen Bruch, der aber offensichtlich nicht zu grösseren Konflikten geführt hat, was auch mit dem Status und dem teilweise autokratischen Führungsstil von Stiefel zusammenhängen mochte (vgl. Läuchli, 1993, 16).

21 Der Begriff „Organe" wurde von John von Neumann im „First Draft" eingeführt und von Rutishauser, Speiser und Stiefel (1951) übernommen. Diese Terminologie war das Ergebnis von Diskussionen, die John von Neumann innerhalb der „cybernetics group" mit Warren McCulloch und Walter Pitts geführt hatte. McCulloch und Pitts hatten 1943 in einer wichtigen Arbeit gezeigt, dass man das Verhalten eines neuronalen Systems in Termini der Aussagenlogik beschreiben kann und dass jeder Algorithmus im Prinzip durch ein solches neuronales Modell – ein McCulloch-Pitts-Netz – realisiert werden kann (vgl. Heims, 1991, 19 ff.; Goldstine, 1972, 274 ff.).

Noch 1950 gab es aus der Sicht von Stiefel zwei mögliche Vorbilder – die IAS-Maschine und die Maschinen von Aiken. Im Gegensatz zur IAS-Maschine verwendete Aiken das Dezimalsystem und setzte die Röhrentechnik nur beschränkt ein. Elektronische Komponenten waren in seinen Augen technisch noch zu wenig zuverlässig. Zudem war er ein entschiedener Gegner der Programmspeicherung. Zu Beginn seiner Recherchen votierte Stiefel noch ganz eindeutig für Aikens Variante. In seinem Reisebericht von 1949 hält er bündig fest: „Der Weg für unser Land scheint mir demnach klar vorgezeichnet. Falls wir etwas tun wollen, bauen wir die von Aiken beschriebene Maschine" (Stiefel, 1949b, 7), und noch 1951 wird die Programmspeicherung bzw. das durch sie ermöglichte „Rechnen mit Befehlen" mit grosser Skepsis betrachtet: „Zusammenfassend sei festgestellt, dass die beiden grundsätzlich verschiedenen Verfahren, mit Adressen zu rechnen, zwar praktisch ebenbürtig sind, doch bietet die Verwendung eines i-Registers (Aikens Alternative, F.F., B.H.) den Vorteil grösserer Rechensicherheit, denn es kann nicht bestritten werden, dass das Rechnen mit Befehlen, die als Zahlen aufgefasst werden, ein gewisses Risiko einschliesst [...] Es ist dies auch der Grund, warum H.H. Aiken Zahlen und Befehle streng getrennt speichert und das Rechnen mit Befehlen ablehnt, solange nicht die Zuverlässigkeit der zur Verwendung kommenden Schaltelemente ganz erheblich gesteigert werden kann" (Rutishauser, Speiser und Stiefel, 1951, 56).[22]

Diese eindeutige Favorisierung der Aiken-Architektur hat Stiefel allerdings schon bald zurückgenommen. 1952, als mit dem Bau der ERMETH begonnen wurde, scheint die Von-Neumann-Architektur das Referenzmodell gewesen zu sein. Stiefel ordnete sich damit in einen Trend ein, der bereits in der erwähnten tabellarischen Übersicht sichtbar wird (Rutishauser, Speiser und Stiefel, 1951, 96 ff.). Vergleicht man die zu dieser Zeit (1949) bereits realisierten mit den in Bau befindlichen Computern, so lässt sich ein deutlicher Trend in Richtung Dualsystem, Fixkomma-Arithmetik, sequentielle Befehlsausführung, paralleles Rechenwerk und gemeinsame Speicherung von Daten und Befehlen feststellen. Diese Entwicklung stützt die in der Computergeschichte verbreitete These, dass die Von-Neumann-Architektur Anfang der 50er Jahre zu einem handlungsleitenden Paradigma wurde[23], und sie entspricht auch der (retrospek-

22 Ein Indiz für das besondere Ansehen Aikens ist auch die Tatsache, dass im Computerraum ein Bild von Aiken hing – und nicht etwa ein Bild des weit berühmteren John von Neumann.

23 Dass die Von-Neumann-Architektur normierend wirkte, zeigt sich auch darin, dass bis in die 70er Jahre parallele Computerarchitekturen kaum Realisierungschancen hatten. Alle Versuche, einen Parallelcomputer zu bauen, scheiterten nicht so sehr an technischen Schwierigkeiten, sondern daran, dass die Von-Neumann-Architektur für die meisten – und insbesondere auch für die Geldgeber – lange Zeit das einzige Denkmodell war. Zu diversen erfolglosen Versuchen, einen Parallelcomputer zu bauen, vgl. Slotnick (1982). Erst in den 70er Jahren begann sich eine Umorientierung abzuzeichen, parallel zu einem ganz ähnlich gelagerten Perspektiven-

tiven) Einschätzung der Beteiligten: hinsichtlich der Architektur habe man sich an von Neumann orientiert, bei der Technik sei dagegen Aiken das Vorbild gewesen. Diese Einschätzung scheint richtig zu sein – nur sagt sie nicht sehr viel aus. Abgesehen von der technischen Ebene, auf der eine Vielzahl von Entscheidungen zu treffen war, gab es auch auf der mathematisch-konzeptionellen Ebene eine Reihe von offenen Problemen, deren Lösung durch die Von-Neumann-Architektur nicht endgültig festgelegt war. Wie wir im folgenden an vier ausgewählten Beispielen zeigen werden, waren viele Entscheidungen, die die ERMETH-Gruppe traf, durch lokale Konstellationen bestimmt. Die Von-Neumann-Architektur hat zwar die Funktion eines Leitbildes gehabt und insofern den groben Rahmen festgelegt. Wie dieser aber konkret gefüllt wurde, war in entscheidenden Masse von den lokalen Bedingungen dieses spezifischen Computerprojektes geprägt.

4.1 Das Zahlensystem: dual oder dezimal?

Bereits 1949 zeichnete sich ein deutlicher Trend in Richtung Dualsystem ab (Rutishauser, Speiser und Stiefel 1951, 96 f.). Während die bereits gebauten Computer vorwiegend dezimale Maschinen waren (6 von 10), verwendeten die im Bau befindlichen Computer mehrheitlich das duale Zahlensystem und folgten damit der von John von Neumann vorgeschlagenen Architektur (6 von 8). Die ERMETH besass dagegen ein dezimales Zahlensystem und stand damit in deutlichem Kontrast zum generellen Trend. Die Entscheidung für das Dezimalsystem ist vor allem auch angesichts der Tatsache erstaunlich, dass die Z4 eine duale Maschine war und man mit ihr mehrheitlich positive Erfahrungen gemacht hatte. Die Nachteile des Dezimalsystems waren schon damals offenkundig. Zum einen wird der Speicher suboptimal genutzt, zum anderen kompliziert sich der Bau des Rechenwerks, da die Schaltungen dualen Charakter haben. Diesen Nachteilen stand aber ein gewichtiger Vorteil gegenüber, zumindest aus der Sicht der ERMETH-Gruppe: auf der Basis des dezimalen Systems liessen sich die Rechnungen sehr viel einfacher kontrollieren.

Zuverlässigkeit war beim Bau der ERMETH ein leitendes Prinzip: „Sicherheit ist wichtiger als Schnelligkeit", wie Stiefel kategorisch festhielt (Stiefel, 1956, 35). Wie aber will man mit Sicherheit wissen, ob die von der Maschine gelieferten Resultate korrekt sind? Angesichts der Neuartigkeit dieser Technik und den vielen ad-hoc-Entscheidungen, die bei ihrem Bau zu treffen waren, war die Frage der Vertrauenssicherung ein entscheidender Punkt. Soziale Ver-

wechsel in der Künstlichen Intelligenz (vgl. Olazaran 1996). Heute ist die Von-Neumann-Architektur nur noch eine Variante unter mehreren. Gemäss der Klassifikation von Flynn (1972) gehört sie zur Klasse der sog. SISD-Architekturen (SISD = Single Instruction stream, Single Data stream).

fahren, wie wir sie aus dem Bereich der Alltagstechnik kennen, genügen in diesem Fall nicht (vgl. dazu Wagner 1994). Es mussten Strategien entwickelt werden, die technischen Rationalitätsmassstäben genügten. Neben Hardware-Kontrollen, die in der Maschine selbst eingebaut waren, spielten auch mathematische Kontrollen eine grosse Rolle.[24] Die Mathematiker im Projekt wollten die von der ERMETH gelieferten Ergebnisse systematisch nachvollziehen können, und für diesen Zweck war eine Dezimaldarstellung sehr viel bequemer als eine duale. Ohne Umrechnungen vornehmen zu müssen, liessen sich die Ergebnisse direkt ablesen. Die Entscheidung, die maschinellen Zahlenoperationen dem gewohnten menschlichen Vorgehen nachzubilden, hatte die Funktion, die internen Abläufe berechenbarer und transparenter zu machen. Diese „Analogisierung" hat damit wesentlich dazu beigetragen, das Vertrauen in diese neue Technik zu vergrössern. Oder wie es ein Beteiligter formulierte: „Wenn schon Fehler, dann die gleichen wie früher."[25]

Wie bereits erwähnt, hat das Dezimalsystem den Nachteil, grössere Anforderungen an das Design des Rechenwerkes zu stellen. Diesen Nachteil versuchte man durch zwei Konstruktionsentscheidungen auszugleichen. Zum einen entschied man sich für eine „single data stream"-Architektur, d. h. das Rechenwerk verarbeitete jede Zahl seriell, so wie Menschen auf Papier addieren. Diese Entscheidung verringerte zwar die Geschwindigkeit, reduzierte aber den Materialaufwand drastisch. Zum anderen wurden die Dezimalziffern intern dual repräsentiert. Obschon also die ERMETH dezimal rechnen konnte, wurden die Dezimalzahlen intern ziffernweise dual repräsentiert und verarbeitet.[26] Die ERMETH war also genau genommen keine reine Dezimalmaschine,

24 Diese mathematischen Kontrollen hatten unter anderem auch die Funktion, den von John von Neumann beschriebenen Zirkel zu stoppen: „It is evident that there is no such thing as perfect checking, because any checking is done by an additional organ, and evidently the additional organ is a new source of errors. Clearly, the problem of checking is renewed at every stage, and it can never be rigorously ended. Whenever you put a set of policemen to watch over the troubles in the machine, you should, on the next level, put a second set of policemen on top of the first one, and so on" (von Neumann, 1946, 272).

25 Fehlerkontrollen wurden nicht erst beim Digitalcomputer eingesetzt, sondern waren auch bei den menschlichen Rechnern üblich. Die Industrie scheint allerdings in die Maschinen mehr Vertrauen gehabt zu haben als in die Frauen, die auch in der Schweiz mehrheitlich die Rechenarbeit besorgten. Die Betriebe mussten jedenfalls immer wieder darauf hingewiesen werden, dass nicht nur Frauen, sondern auch Maschinen mitunter Fehler begehen. Zur Organisation der menschlichen „Computer" vgl. Ceruzzi 1991 sowie Comrie, der noch 1944 den jungen Frauen den Rechenberuf als Karrieremöglichkeit empfahl (Comrie 1944).

26 Um eine Dezimalzahl binär darzustellen, braucht man mindestens 4 Binärstellen. Auf diese Weise sind im Prinzip 16 Zahlen darstellbar (24 = 16). Da in einer Dezimalmaschine nur die Zahlen von 0 bis 9 dargestellt werden müssen, bleiben sechs Kombinationen ungenutzt. Wenn man z. B. die Zahl 128 mit dem Aiken-Verfahren verschlüsselt, erhält man 0001 0010 1110. Die auf diese Weise kodierte Dezimalzahl kann dann ziffernweise verarbeitet werden. Bei der Vercodung orientierte sich die ERMETH-Gruppe an einem von Aiken entwickelten Verfahren (Rutishauser, Speiser und Stiefel, 1951, 22). Technische Überlegungen haben bei dieser Wahl

sondern eine Binärdezimalmaschine. Dies ermöglichte auf der Hardware-Ebene eine beträchtliche Vereinfachung des Rechenwerks.

4.2 Die Arithmetik: Fixkomma oder Gleitkomma?

Die Arithmetik der ERMETH ist ein gutes Beispiel für ein eigenwilliges Design. Während die ERMETH auf technischem Gebiet sehr konservativ war, war sie auf der mathematisch-konzeptionellen Ebene ausgesprochen innovativ. Die erstaunlichste Entscheidung ist die Entscheidung für eine Gleitkomma-Darstellung. Von den damals bekannten Computern verwenden zwar drei ein Gleitkomma (Z4, Model V, Mark II), es handelt sich jedoch bei allen um Relais-Computer. Bei elektronischen Computern war eine Gleitkomma-Darstellung sehr viel schwieriger zu implementieren. Maurice Wilkes, der den ersten Computer mit einer Von-Neumann-Architektur gebaut hatte, stellt denn auch unmissverständlich fest: „That the EDSAC would be a fixed-point maschine went without saying. [...] It would be a long time before any electronics designer would be prepared to attempt such a thing" (Wilkes, 1985, 429). Das war 1949. Bis zum Bau der ERMETH änderten sich die technologischen Randbedingungen kaum. Dennoch, und in deutlichem Kontrast zum herrschenden Trend, entschied sich das ERMETH-Team für ein Design mit Fix- *und* Gleitkomma-Darstellung.[27] Welche Überlegungen führten zu dieser Entscheidung?

In einer digitalen Maschine können Zahlen nur durch endliche Ziffernfolgen dargestellt werden (vgl. zum folgenden Rutishauser, Stiefel und Speiser, 1951, 25 ff.). Im Gegensatz zur Gleitkomma- wird bei einer Fixkomma-Darstellung dem Komma eine feste Position zugewiesen. Ein solcher Computer rechnet im wesentlichen mit den Regeln der ganzzahligen Arithmetik.[28] Die frühen Designer waren der Meinung, dass der entscheidendste Faktor für die Zuverlässigkeit komplexer Berechnungen die Genauigkeit der Zahlendarstellung, d. h. die Anzahl der Stellen hinter dem Komma war. Frühe Computer setzten

offenbar nur eine sekundäre Rolle gespielt. Ausschlaggebend scheint das Ansehen gewesen zu sein, das Aiken bei Stiefel genoss.

27 Aus diesem Grund hatte die ERMETH auch einen doppelten Befehlssatz, einen für Festkomma- und einen für Gleitkomma-Zahlen. Dies hatte zur Folge, dass ihr Befehlssatz vergleichsweise umfangreich war: er bestand aus 49 Instruktionen. Die EDSAC von Maurice Wilkes kam dagegen mit einem Dutzend Befehle aus.

28 Aus der Sicht des Rechners spielt es keine Rolle, ob 1,234 und 2 oder 1234 und 2 multipliziert wird. Das Resultat ist entweder 2,468 oder 2468, je nachdem, wie der Programmierer das Resultat interpretiert.

das Komma entweder vor oder nach der ersten Stelle. Ihre Genauigkeit belief sich auf etwa 12 Dezimal- bzw. 40 Dualstellen nach dem Komma.[29]

Die Fixkomma-Darstellung hat jedoch einen deutlichen Nachteil. Die Benutzer mussten bei jeder Rechnung überschlagsmässig nach- (bzw. voraus-)rechnen, ob die Resultate noch in den zulässigen Zahlenbereich fielen. Nötigenfalls musste das Problem neu „skaliert" werden. Die Einheiten der Variablen wurden dann so gewählt, dass die zugehörigen Werte in den zulässigen Zahlenbereich zu liegen kamen.[30] Der Umgang mit Fixkomma-Darstellungen setzt somit eine relativ hohe Expertise im Bereich des numerischen Rechnens voraus. Bei einer Gleitkomma-Darstellung stellt sich dieses Problem nicht. Dabei wird normalerweise die sog. halblogarithmische Darstellung gewählt, die erstmals wohl von Konrad Zuse benutzt wurde. Eine Zahl Z wird intern durch eine Mantisse m und einen Exponenten e dargestellt. Z wird aus Mantisse und Exponent wie folgt berechnet: $Z = m \cdot b^e$, wobei b die Basis bezeichnet. Unter der Bedingung, dass der Wertebereich des Exponenten genügend gross gewählt ist, erlaubt diese Darstellung (endliche) Zahlen praktisch beliebiger Grössenordnung zu bearbeiten und ist somit für den Benutzer sehr viel einfacher zu handhaben.[31]

Die Grössenordnung des Wertebereiches des Exponenten war also der entscheidende Punkt. „Der Wertbereich", so Stock in seiner Beschreibung der ERMETH-Architektur, „soll hinreichend gross sein, so dass beim Programmieren auf die Grössenordnung der Zahlen keine Rücksicht genommen wer-

29 Um nur einige Beispiele zu erwähnen: Die EDSAC von Maurice Wilkes rechnete mit 36 Dualstellen, die IAS-Maschine mit 40 und der ACE-Computer von Alan Turing mit 32 (Lavington,1980, 118 ff.).

30 Angenommen, der Wert 1,234 ist eine Volumenmessung, etwa Deziliter. Dann könnte der Programmierer die Messung in Liter ausdrücken. Die Umrechnung ergibt 0,123. Aufgrund der begrenzten Genauigkeit geht bei der Umwandlung eine Stelle verloren.

31 Diese Überlegung spielte für John von Neumann und seine Mitarbeiter offensichtlich keine Rolle. In ihrer „Preliminary Discussion" führen sie eine Reihe von Gründen an, weshalb die Gleitkommadarstellung abzulehnen sei: „Several of the digital computers being built or planned in this country and England are so-called ‚floating decimal point'. [...] There appear to be two major purposes in a ‚floating' decimal point system [...] The first of these purposes is to retain in a sum or product as many significant digits as possible and the second of these is to free the human operator from the burden of estimating and inserting into a problem ‚scale factors' – multiplicative constants which serve to keep numbers within the limits of the machine. There is, of course, no denying the fact that human time is consumed in arranging for the introduction of suitable scale factors. We only argue that the time so consumed is a very small percentage of the total amount of time we will spend in preparing an interesting problem for our machine. The first advantage of the floating point is, we feel, somewhat illusory. In order to have such a floating point one must waste memory capacity which could otherwise be used for carrying more digits per word. It would therefore seem to us not at all clear whether the modest advantages of a floating binary point offset the loss of memory capacity and the increased complexity of the arithmetic and control circuits" (Burks, Goldstine und von Neumann, 1946, 106).

den muss" (Stock, 1956, 9). Während der Exponent in einem ersten Entwurf noch auf ± 31 beschränkt war, wurde er in der endgültigen Variante auf ± 200 erweitert. Diese drastische Erweiterung des Wertebereichs des Exponenten ging auf Heinz Rutishauser zurück, der als Mathematiker die Architektur der ERMETH entscheidend prägte.[32]

Damit unterschied sich die ERMETH wesentlich von anderen Computern mit Gleitkomma-Darstellung, deren Wertebereich sehr viel niedriger lag. Rutishauser war weltweit praktisch der einzige, der die Bedeutung eines hohen Wertebereichs für die Zuverlässigkeit der numerischen Berechnungen erkannt hatte.[33] In den anderen Computerprojekten wurde Zuverlässigkeit primär als ein Hardware-Problem angesehen. Nicht nur die Gleitkomma-Lösung, sondern die Arithmetik allgemein sei, das ist das einhellige Urteil, bei der ERMETH ausserordentlich durchdacht und zukunftsweisend gewesen. Es habe Jahre gedauert, bis ein kommerzieller Computer eine ähnliche „saubere" Arithmetik aufgewiesen habe. Programme, die auf der ERMETH einwandfrei gelaufen waren, zeigten teilweise erhebliche Probleme, wenn sie später auf kommerziellen Maschinen implementiert wurden.[34]

Die Entscheidung, entgegen dem herrschenden Trend und trotz erheblicher technischer Schwierigkeiten, eine Gleitkomma-Darstellung zu verwenden, hat mit drei für das ERMETH-Projekt spezifischen lokalen Bedingungen zu tun. Die ERMETH hatte eine Doppelfunktion: zum einen sollte sie ein Werkzeug sein für die wissenschaftliche Forschung im Bereich der numerischen und angewandten Mathematik, zum anderen hatte sie aber auch Dienstleistungsfunktion sowohl für die ETH wie auch für externe Kunden.[35] Um das Personal am *Institut für angewandte Mathematik* nicht über Gebühr mit der „langweiligen Arbeit des Programmierens" (Stiefel) zu belasten, wurde eine Maschine konzipiert, deren Bedienung und Programmierung so einfach war, dass sie nach kurzer Einarbeitungszeit von den Kunden selbständig benutzt werden konnte. Wir kommen auf diesen Benutzeraspekt im folgenden Abschnitt zu-

32 Zu Rutishauser vgl. u. a. Bauer (1980); Läuchli, 1993, 14.

33 Wie Stock in seiner Abhandlung anmerkt, wurde die Relevanz des Exponenten bis zu dieser Zeit lediglich an zwei Stellen erkannt – in einer Dissertation in Harvard (Coolidge 1953) und in einem Vortrag anlässlich einer Computertagung in Harvard 1951 (Lester 1951). Nur in München scheint man die Bedeutung des Exponenten realisiert zu haben (Piloty und Piloty 1953).

34 Auf die Frage, wie die Gleitkomma-Darstellung technisch umgesetzt wurde, gehen wir hier nicht ein, vgl. dazu Stock (1956).

35 Aus einer Zusammenstellung aus dem Jahre 1960 geht hervor, dass die ERMETH in diesem Jahr pro Monat durchschnittlich 60 Stunden für das *Institut angewandte Mathematik* eingesetzt wurde, 70 Stunden für andere ETH-Institute und 40 Stunden für externe Kunden. Es ist anzunehmen, dass in den ersten Betriebsjahren der Einsatz für die externen Kunden grösser war.

rück. Ein weiterer wesentlicher Faktor waren die Erfahrungen, die man mit der Z4 gesammelt hatte, die ebenfalls eine Gleitkomma-Arithemetik verwendete. Abgesehen von Aiken und Stibitz waren Stiefel und seine Mitarbeiter weltweit vermutlich die einzigen, die mit einer Gleitkomma-Arithmetik praktische Erfahrungen gesammelt hatten. Ein dritter Faktor schliesslich war die personelle Zusammensetzung des Teams. Etwas überspitzt formuliert: die ERMETH wurde in einem mathematischen Institut für mathematische Zwecke von Mathematikern gebaut. Diese spezifisch mathematische Ausrichtung hat das Design entscheidend geprägt.

4.3 Benutzerfreundlichkeit

Ein wesentliches Prinzip, das den Bau der ERMETH leitete, war das Prinzip der Benutzerfreundlichkeit. Das erste Postulat, so Stiefel, bestehe darin, „dieses Gerät so einfach und übersichtlich zu organisieren, dass die Mühen der Programmierung auf ein Mindestmass herabgedrückt werden" (Stiefel, 1957, 509). Dies war zu dieser Zeit eine aussergewöhnliche Perspektive. Bei den anderen Entwicklungsprojekten standen ganz andere Ziele im Vordergrund – Schnelligkeit, technische und konzeptionelle Innovativität –, aber ganz gewiss nicht die leichte Bedienbarkeit durch den (externen) Benutzer.[36] Die Bedeutung, die der Benutzerfreundlichkeit zugemessen wurde, hat mit den spezifischen Bedingungen zu tun, unter denen die ERMETH entwickelt und eingesetzt wurde. Neben ihrer wissenschaftlichen Funktion hatte die ERMETH, wie bereits erwähnt, vor allem auch Dienstleistungsaufgaben zu erfüllen.

Wie Stiefel an verschiedenen Stellen betont, war die damit verbundene Programmierarbeit enorm, zumal die ERMETH für mehrere Jahre in der Schweiz der einzige Computer war und Routineaufträge selten waren, sehr viel seltener jedenfalls als in grösseren Ländern, wo es häufiger vorkam, dass verschiedene Firmen dasselbe Problem zu berechnen wünschten. Anstatt das gleiche Programm für verschiedene Aufträge benützen zu können, wechselten die Probleme häufig, und für jedes Problem musste ein neues Programm geschrieben werden. Die „Externalisierung" der Programmierarbeit war in den Augen von Stiefel die einzige Lösung, um zu verhindern, dass sein Institut zu einem reinen Dienstleistungsbetrieb verkam und seiner eigentlichen Aufgabe nicht mehr gerecht wurde. Dies bedingte aber, dass die Maschine relativ leicht zu

36 Dass es sich bei der Benutzerfreundlichkeit der ERMETH keineswegs um eine selbstverständliche Anforderung handelt, macht z. B. auch die Haltung Alan Turings deutlich: „Williams took great care that Turing did not meddle in the engineering design of the machine that he and Kilburn were building, but he did let him design the programming system, and this, as anyone knowing Turing would expect, made few concessions to the comfort of the user" (Wilkes, 1985, 138).

bedienen und zu programmieren war. Die Gefahr, dass die Mitarbeiter des Institutes nicht mehr für die eigentliche mathematische Arbeit zur Verfügung stehen, sondern „durch die Maschine dauernd gehetzt werden", könne nur dann vermieden werden, „wenn die Organisation des Automaten so klar und einfach ist, dass das Programmieren für den von aussen kommenden Ingenieur nicht als eine komplizierte Meta-Mathematik erscheint, die er ablehnt, und zwar mit Recht" (Stiefel, 1956, 30). Genau darauf ist das Design der ERMETH ausgerichtet. „Nichts ist unsachlicher", so Stiefel 1957, „als einen Rechenautomaten nur nach der Geschwindigkeit beurteilen zu wollen, mit der er etwa eine Addition ausführt. Unsere Erfahrungen seit 1950 im automatischen Rechnen zeigen vielmehr, dass Einfachheit und Klarheit der Bedienung und vor allem einfache Programmierbarkeit oft viel wichtiger ist" (Stiefel, 1957, 509).[37]

Viele Design-Entscheidungen der ERMETH sind eine Folge dieser Grundüberlegung, allen voran die Entscheidung für das Gleitkomma und die Dezimaldarstellung. Beides hat die Arbeit mit der ERMETH wesentlich erleichtert. In einem 1956 verfassten Aufsatz erwähnt Stiefel noch eine Reihe von anderen Design-Merkmalen, die vom Prinzip der Benutzerfreundlichkeit geleitet waren. Dazu gehört z. B. die Entscheidung, nur einen Speicher, und nicht einen Primär- und einen Sekundärspeicher, zu benutzen, da jeder zusätzliche Speicher das Programmieren kompliziere, oder die Möglichkeit, die Befehle in ihrer ursprünglichen mathematischen Bezeichnung einzugeben, und nicht, wie sonst üblich in verschlüsselter Form (Stiefel, 1956, 31 f.).

Dies mag aus der Perspektive einer Windows-gewohnten Benutzerin nicht besonders radikal erscheinen, zur damaligen Zeit wurde aber mehrheitlich noch in Maschinensprache programmiert, d. h. Programme bestanden aus langen Reihen von Nullen und Einsen. Um die Programmierarbeit etwas zu erleichtern, hat Maurice Wilkes für die Programmierung seiner EDSAC alphabetische Abkürzungen benutzt, die dann von einem Spezialgerät in den Maschinencode „übersetzt" wurden.[38] Dies war bei der ERMETH überflüssig. Ihre Programmierung war so einfach und bequem, dass einige der damaligen Benutzer die Entwicklung einer (höheren) Programmiersprache lange Zeit nicht für nötig hielten.

37 Vgl. auch das 1957 herausgebene Benutzerblatt, in dem zwischen drei Kategorien von Problemen unterschieden wird – Probleme aus Naturwissenschaft und Technik, für die zuerst noch eine mathematische Formulierung und ein numerischer Lösungsweg gefunden werden muss; mathematische Probleme, deren mathematische Lösung bekannt ist; und schliesslich Standardprobleme, für die bereits ein Programm besteht –, und in dem die Kunden explizit darauf hingewiesen werden, dass jene Benutzer Priorität haben, für deren Probleme es bereits Programme gibt oder die in der Lage sind, diese selbst herzustellen.

38 Es handelte sich dabei allerdings noch nicht um eine Assemblersprache.

Dass der Benutzerfreundlichkeit eine so hohe Priorität eingeräumt wurde, hatte auch mit den Erfahrungen zu tun, die man mit der Z4 gemacht hatte. Oder wie es der Technik-Verantwortliche formuliert: „Wir wussten, was es bedeutet, eine Maschine in Betrieb zu halten und sie bedienerfreundlich zu machen und wie die Mensch-Maschine-Oberfläche aussehen muss. Die Z4 hat einen starken Einfluss gehabt."[39] Das benutzerfreundliche Design der ERMETH war allerdings nur *eine* Antwort auf das Problem, die Programmierarbeit möglichst einfach zu gestalten. Daneben wurden am *Institut für angewandte Mathematik* noch eine Reihe von anderen „flankierenden" Massnahmen ergriffen. Neben der für europäische Verhältnisse ausgesprochen frühen Einrichtung von ständigen Programmierkursen haben insbesondere Stiefel und Rutishauser sehr bald realisiert, dass nicht die Hardware der entscheidende Punkt ist, sondern die „Software" (die es zu dieser Zeit im heutigen Sinn allerdings noch gar nicht gab). Um die Programmierarbeit zu erleichtern, musste mit anderen Worten beim Programmieren selbst angesetzt werden. Dazu gehörte die Entwicklung von höheren Programmiersprachen[40], die von Stiefel schon sehr früh geforderte internationale Standardisierung der „Terminologie des Programmierens" (Stiefel, 1956, 32) und schliesslich die, wie man es damals nannte, „automatische Programmierung". Damit war der Compilerbau gemeint, zu dem Rutishauser entscheidend beigetragen hat (Knuth und Pardo, 1980, 218 ff.). Das Endziel dieser „bedeutenden Entwicklung" bestehe darin, so Stiefel, „dass die Maschine ihr Programm selbst berechnet, ausgehend von den mathematischen Formeln, die auf Lochkarten geschrieben sind" (Stiefel, 1957, 509).

Die Erkenntnis, dass das Ergebnis einer Berechnung nicht immer eine Zahl zu sein hat, sondern auch eine Befehlsfolge sein kann, setzt voraus, dass man den Computer nicht mehr bloss als eine gigantische Rechenmaschine sieht, sondern sehr viel abstrakter als eine *symbol*verarbeitende Maschine. Durch die grundlagentheoretische Arbeit von Alan Turing hätte eine solche Deutung zwar schon vorgelegen (vgl. dazu Heintz 1993b), aber mit Ausnahme von John von Neumann schien niemand Turings mathematische Arbeit gekannt zu haben, und auch seine eigene am *National Physical Laboratory* entwickelte ACE-Maschine wurde kaum rezipiert. Bis weit in die fünfziger Jahre herrschte eine Deutung des Computers vor, wie sie Aiken exemplarisch zum Ausdruck brachte: „The output of computation exists of numbers only. It has been proposed – facetiously by some, in earnest by others – to build a ‚thinking' machine whose

39 Die Benutzerorientierung der ERMETH war keine isolierte Episode. Die Forderung nach einer übersichtlichen Mensch-Maschine-Schnittstelle und nach einfacher Programmierung stand viele Jahre später im Mittelpunkt einer weiteren Computerentwicklung an der ETH, der von Niklaus Wirth entwickelten LILITH, vgl. dazu Furger (1992).

40 Das *Institut für angewandte Mathematik* war massgeblich an der Entwicklung von ALGOL beteiligt, vgl. zum ALGOL-Projekt Naur (1981).

output would be orders rather than numbers. From a small input such a machine would automatically produce the system of orders required for the solution of a complex problem of a computing machine. Perhaps such a machine will be developed in the future, but all existing machines produce only numbers".

Was für Aiken noch unvorstellbar war, daran hat Heinz Rutishauser seit 1951 gearbeitet. Seine frühen Arbeiten zum „automatischen Programmieren" gelten heute als wichtiger Meilenstein in der Geschichte der Programmiersprachen (Knuth und Pardo 1980). Rutishauser hat offenbar bereits Anfang der 50er Jahre erkannt, dass man den Computer nicht nur für numerische, sondern auch für symbolische Berechnungen einsetzen kann – als Maschine, die „Rechenpläne berechnet" (vgl. Rutishauser 1951; Rutishauser 1952; Bauer 1980). Bereits Anfang der 50er Jahre modifizierte er die Z4 so, dass man sie auch als Compiler einsetzen konnte (Schwarz, 1981, 125), und aus dem ersten Entwurf des ERMETH wird deutlich, dass für Rutishauser ihre Funktion als Compiler im Vordergrund stand (Rutishauser, 1952, Kap. 2 und 3). Dass Rutishauser relativ früh zu einer abstrakten Konzeption des Computers gelangte, hatte zum einen mit seiner mathematischen Ausbildung zu tun, zum anderen aber auch mit der Z4 und dem Kontakt zu Zuse. Zuse hatte bereits einige Jahre zuvor eine Programmiersprache entwickelt, den „Plankalkül", der im Gegensatz zu den zehn Jahre später entstandenen Programmiersprachen wie FORTRAN, ALGOL oder COBOL nicht spezifisch auf numerische Probleme zugeschnitten war, sondern mit dem sich auch „symbolische" Aufgaben lösen liessen (vgl. Zuse, 1984, 190 ff.; Petzold, 1985, Kap. 5). Rutishauser hat Zuses Plankalkül offensichtlich gekannt und gehörte damit zu Beginn der 50er Jahre zu den wenigen Leuten, die mit der Möglichkeit höherer Programmierung praktisch konfrontiert waren.

4.4 Die Basistechnologie: Relais oder Röhren?

Die letzte Einscheidungskonstellation, auf die wir eingehen möchten, liegt eher auf der technischen Ebene und hat nur mittelbar mit der Architektur der ERMETH zu tun. Es handelt sich um die Frage, welche Basistechnologie zur Umsetzung des logischen Designs verwendet werden soll. Hier standen im wesentlichen zwei Technologien zur Verfügung: die Relais- und die Röhrentechnik.[41] Bei beiden Technologien handelte es sich um bewährte Technologien aus dem Bereich der Kommunikationstechnik, die an der ETH zum Lehr-

41 Aus der Sicht einiger Ingenieure wäre es interessanter gewesen, Transistoren zu benutzen. Der Transistor wurde 1947 entwickelt, das ERMETH-Projekt erst 1953 begonnen. Speiser meint allerdings, dass die ERMETH nicht mit Transitoren hätte gebaut werden können (Speiser, 1979, 360), ausserdem hatte sich Stiefel kategorisch dagegen ausgesprochen.

stoff des Studienganges Elektrotechnik gehörten. Beide Technologien wurden seit Jahrzehnten in der Industrie benutzt (vgl. Petzold 1987). Mit der Relaistechnik hatte das ERMETH-Team bereits im Zusammenhang mit der Z4 Erfahrungen gesammelt, die Röhrentechnik wurde von den Ingenieuren zwar im linearen Bereich gemeistert, nicht jedoch im dem für das Design von digitalen Bausteinen wesentlichen nicht-linearen Bereich. Die Erfahrungen im Ausland zeigten, dass die Röhrentechnik zum Bau von digitalen Bausteinen oder ganzen Computern zwar erfolgreich benutzt werden konnte, die Designer jedoch vor beträchtliche Zuverlässigkeitsprobleme stellte.[42] Probleme warf vor allem der Speicher auf, weniger das Rechenwerk.[43]

Aus der Sicht der ERMETH-Gruppe konnte es also durchaus riskant erscheinen, sich auf die Röhrentechnik zu verlassen, zumal eine solche Entscheidung aller Voraussicht nach irreversibel gewesen wäre und Howard Aiken, die grosse Bezugsfigur von Stiefel und teilweise auch von Speiser, ein entschiedener Gegner der Röhrentechnik war. Mit seiner Aversion gegen die Röhrentechnik schien Aiken zu dieser Zeit auch recht gehabt zu haben: während seine Computer zufriedenstellend funktionierten, hatten der EDVAC und der IAS-Computer von John von Neumann mit erheblichen technischen Problemen zu kämpfen.[44] Dies hat auch die ERMETH-Gruppe beeinflusst. In einem Bericht über seine zweite USA-Reise 1951/52 hält Stiefel jedenfalls fest, dass eine „heilsame Ernüchterung hinsichtlich der Ansichten über die superschnellen elektronischen Computer eingetreten" sei (Stiefel, 1952, 4). Und in dem zusammen mit Rutishauser und Speiser verfassten Überblick über den gegenwärtigen Stand der Computertechnik wird ganz eindeutig die Relaistechnik favorisiert (vgl. z. B. S. 10).

Es ist deshalb nur folgerichtig, dass die ERMETH-Gruppe 1951/52 entschied, einen Relaiscomputer zu bauen. Diese Entscheidung wurde allerdings

42 Zu jener Zeit galt ein System mit zwei- oder dreihundert Röhren als durchaus komplex. Der einfachste Computer benötigte bereits etwa tausend, die monströse ENIAC sogar 18'000 Röhren.

43 Dies war jedenfalls die Auffassung von Stiefel in seinem Reisebericht von 1952: „Die Schwierigkeiten liegen nicht bei den elektronischen Rechenwerken – die ENIAC arbeitet seit 1946 anstandslos mit einem solchen –, sondern bei den hohen Anforderungen an die Speicherwerke (‚Gedächtnis der Maschine')". Stiefels Beobachtung wird auch von anderen bestätigt. So konnte für den IAS-Computer die ursprünglich vorgesehene, durch RCA herzustellende Selectron-Röhre nicht benutzt werden (Bigelow 1980). Auch die Benutzung von „mercury-delay-lines" und Magnet-Trommel war alles andere als unproblematisch.

44 Die technischen Probleme des IAS-Computers waren allerdings nicht auf die Röhren beschränkt. Sowohl die Williams-Röhren, die für den Speicher des IAS-Computers benutzt wurden, als auch die „mercury delay lines" der EDSAC stellten die damaligen Wissenschafter vor erhebliche betriebliche Probleme Für eine Diskussion technologischer Probleme beim Baum des IAS-Computers siehe Bigelow (1980). Zum Bau des EDSAC siehe Wilkes (1985).

bereits 1953 revidiert, ohne dass im Jahresbericht Gründe für dieses Umdenken angegeben werden (Jahresbericht 1953). Während Ambros Speiser, der damalige Leiter des technischen Bereichs, diese Umorientierung auf zunehmende Erfahrung und Sicherheit zurückführt, stellt John Richard Stock technische Schwierigkeiten in den Vordergrund (Stock, 1956, 9). Ob die Entscheidung, auf die Röhrentechnologie umzusteigen, durch technische Schwierigkeiten oder durch wachsende technische Kompetenz und Entscheidungsautonomie erklärbar ist, lässt sich heute nicht mehr mit Sicherheit rekonstruieren. Zudem schliessen sich die beiden Erklärungen gegenseitig auch nicht aus. So ist denkbar, dass anfänglich auftretende technische Probleme dazu beitrugen, die getroffene Entscheidung in Frage zu stellen. Gleichzeitig mag die wachsende technische Kompetenz dazu geführt haben, dass manche stillschweigend akzeptierten technischen „Fakten" und Erfahrungsregeln relativiert wurden.

In den retrospektiven Darstellungen wird diese Episode in der Regel nicht erwähnt. Die Geschichte der ERMETH wird stattdessen als eine gradlinige und technisch notwendige Entwicklung dargestellt, als ein Weg, zu dem es keine Alternativen gab. Die Tatsache, dass, wie das Beispiel der Relais-/Röhrentechnik zeigt, Entwicklungswege abgebrochen wurden und die getroffenen Entscheidungen keineswegs immer einer technischen Sachlogik gehorchten, sondern oft durch partikuläre Konstellationen beeinflusst waren, wird zum grossen Teil ausgeblendet. Die ERMETH-Geschichte ist eine Erfolgsgeschichte – dies soll nicht bestritten werden. Zu diesem Erfolg gehört aber gerade das, was von den Beteiligten in der Regel nicht thematisiert wird – das Abweichen von der (technischen) Regel, die geschickte Nutzung lokaler Ressourcen, die Improvisation.

5. Schlussfolgerungen

Zum Schluss möchten noch einmal auf die eingangs formulierten techniksoziologischen Perspektiven zurückkommen und am Beispiel der ERMETH-Fallstudie die von uns kritisierten Punkte zusammenfassend rekapitulieren.

(1) *Kontingenz vs. Determinismus*. Wie die Geschichte der ERMETH beispielhaft zeigt, ist auch nach der Durchsetzung eines technologischen Paradigmas noch längst nicht alles festgelegt. Die Von-Neumann-Architektur legt nicht einen Konstruktionsalgorithmus fest, sondern definiert einige wenige grundlegende Prinzipien, die einen weiten Handlungsspielraum offen lassen. Im Gegensatz zum sozialkonstruktivistischen Modell, das Kontingenz bis zum Moment der Durchsetzung eines technologischen Paradigmas behauptet, die daran anschliessenden Prozesse aber als weitgehend determiniert ansieht, betonen

wir die *Offenheit* des technischen Handelns auch nach der Phase der „Schliessung".

Die in der Computergeschichte verbreitete Annahme, dass die Von-Neumann-Architektur den Status eines technologischen Paradigmas hatte, unterstellt zudem eine begriffliche Eindeutigkeit, die es in dieser Form damals nicht gab. Die Berichte, die John von Neumann und seine Mitarbeiter schrieben, erläutern detailliert die funktionale Organisation und die technische Realisierung der Von-Neumann-Maschine. Aus dieser Fülle von Informationen jene zu destillieren, die heute als zentrale Komponenten einer „Von-Neumann-Architektur" angesehen werden, hätte eine Distanzierungs- und Abstraktionsleistung erfordert, die damals kaum möglich war. „Von-Neumann-Maschine" konnte zu dieser Zeit verschiedenes bedeuten: die Maschine, die von Neumann in seinem „First Draft" beschrieben hatte und deren Vorbild der EDVAC war, oder die IAS-Maschine, deren Design in der „Preliminary Discussion" erläutert wurde. Die beiden Maschinen unterschieden sich in verschiedener Hinsicht – sie verwendeten einen unterschiedlichen Befehlssatz, hatten ein anderes Adress-System[45] und unterschieden sich auch in bezug auf das Rechenwerk: der EDVAC hatte ein serielles Rechenwerk, der IAS-Computer ein paralleles. Auch auf der technischen Ebene gab es eine Reihe von Unterschieden: der EDVAC-Computer benutzte z. B. als Speicher „mercury delay lines", der IAS-Computer eine Variante der Williams-Röhren. Mit „Von-Neumann-Maschine" konnte zudem die technische Realisierung gemeint sein oder aber die abstrakte funktionale Organisation – ihre Architektur. Heute ist es üblich, den Computer in Termini eines Schichtenmodells zu beschreiben (vgl. u. a. Giloi, 1981), und wir haben gelernt, dass es möglich ist, von den nächstunteren Schichten zu abstrahieren. Den Anfang zu dieser Sichtweise hat von Neumann gemacht, als er in seinem „First Draft" den EDVAC nicht anhand technischer Eckdaten, sondern in Termini seiner funktionalen Organisation beschrieb (vgl. Smith, 1989, 280). Obschon diese Abstraktion eine zentrale Leistung John von Neumanns war, indem er, wie es Zuse formulierte, „aus der Fülle der Möglichkeiten das herauspickte, was wirklich wichtig war" (Zuse, 1992, 13), erscheint die „Von-Neumann-Architektur" in diesen beiden Berichten noch nicht als festumrissener Begriff mit einer Reihe von aufzählbaren Definitionselementen. Dazu ist es erst später gekommen – wie man in der Retrospektive erkennt.

45 Der EDVAC-Computer war eine Vier-Adress-, der IAS-Computer eine Ein-Adress-Maschine. Bei einer Vier-Adress-Maschine besteht jeder Befehl aus einem Operationscode, d. h. einer Ziffernfolge, die festlegt, welche Operation durchgeführt werden soll, und vier Adressen. Die ersten zwei Adressen identifizieren Plätze im Speicher, aus denen die Operanden geholt werden sollen, die dritte legt den Platz, an dem das Ergebnis deponiert werden soll, fest, und die vierte und letzte Adresse fixiert den nächsten Operationsschritt.

Damit ist ein weiterer Grund angesprochen, weshalb technologische Paradigmen nicht die Rolle spielen, die ihnen in der techniksoziologischen Literatur zugewiesen wird. Paradigmen (und dies gilt für wissenschaftliche wie für technologische Paradigmen) werden in der Regel als gedankliche Modelle interpretiert, die dem Handeln vorausgehen und es anleiten – als Handlungspläne gewissermassen. Im Gegensatz zu dieser Sicht möchten wir die These aufstellen, dass (technologische) Paradigmen ihre Bedeutung erst im nachhinein erhalten – als retrospektive Erklärungen situativ bedingter Entscheidungen. Für Stiefel und seine Mitarbeiter war die Von-Neumann-Architektur keine klar abgegrenzte Zielvorgabe. Dazu ist sie erst im nachhinein geworden. Erst von heute aus gesehen bekommt die Von-Neumann-Architektur ihre Funktion: sie dient dazu, die vielen situativ und ad-hoc getroffenen Entscheidungen zu linearisieren und ihnen eine Plausibilitätsstruktur zu unterlegen, ganz ähnlich wie es die Wissenschaftssoziologie für die wissenschaftliche Praxis beschreibt (vgl. u. a. Knorr-Cetina, 1984, Kap. 5 und 6; Gooding, 1992, 76 ff.).[46]

Die wissenschaftssoziologische Unterscheidung zwischen dem faktischen Forschungshandeln und seiner nachträglichen Rekonstruktion (z. B. in Form einer Publikation) orientiert sich an Unterscheidung zwischen Handlungsverlauf und Handlungserklärung, wie sie vor allem in der Handlungstheorie G.H. Meads und in der Ethnomethodologie zu finden ist. Lucy Suchman, die die pragmatische Handlungstheorie für die Techniksoziologie fruchtbar zu machen versucht, unterscheidet in diesem Zusammenhang zwischen „situated actions" und „plans" (bzw. zwischen pragmatischen und teleologischen Handlungstheorien, die Handlungen nach dem Zweck/Mittel-Schema konzipieren). Während teleologische Handlungstheorien die Beziehung zwischen Plan und Handlung als Kausalbeziehung begreifen, sind „Pläne" (oder Paradigmen) aus der Sicht einer pragmatischen Handlungstheorie nachträgliche „Rationalisierungen" (im Sinne von Erklärungen und Rechtfertigungen): „In our everyday action descriptions we do not normally distinguish between accounts of action provided before and after the fact, and the action's actual course. As common sense constructs, plans are a constituent of practical action, but they are constituent as an artifact of our *reasoning about* actions, not as the generative *mechanism of* action" (Suchman, 1987, 38 f.). Diese Einsicht gilt auch für die Techniksoziologie und stützt von theoretischer Seite her unsere These, dass technologische Paradigmen weniger Handlungsanleitungen sind, sondern oft den Status von nachgeschobenen Erklärungen haben. Pointiert formuliert: tech-

46 Dies wurde in vielen Interviews deutlich, in denen Fragen nach den Gründen für eine bestimmte Entscheidung mit dem Hinweis beantwortet wurde: „Weil das bei einer Von-Neumann-Architektur so ist". Gleichzeitig, und im Widerspruch dazu, wurde mehrfach betont, dass es damals sehr schwierig gewesen sei, sich zu orientieren. Man habe noch keine sicheren Beurteilungskriterien an der Hand gehabt.

nisches Handeln folgt nicht technischen Paradigmen, sondern Paradigmen sind nachträgliche Systematisierungen situativer und kontingenter Praktiken.

Diese Bemerkung ist natürlich auch als ein methodisches Caveat zu verstehen. Die Darstellungen, die unsere Interviewpartner von der Entwicklung der ERMETH gaben, sind nicht Beschreibungen des tatsächlichen Verlaufes, sondern Rekonstruktionen, durch die eine Vielzahl von komplexen Situationen *nachträglich* in ein lineares Schema – in einen nachvollziehbaren und logischen Ablauf gebracht werden, der von Zufälligkeiten und lokalen Besonderheiten „gereinigt" ist. Die Von-Neumann-Architektur bietet sich hier als ein vorgefertigtes und breit akzeptiertes Schema an, das dazu dient, den vielen punktuellen Entscheidungen eine Struktur zu geben und sie nachträglich zu rechtfertigen.

(2) *Kontextualität vs. Universalität.* Paradigmen, so unsere erste These, gehen in der Regel den Praktiken nicht voraus, sondern haben den Status von nachträglichen Systematisierungen. Zudem definieren sie nur einige wenige allgemeine Prinzipien, die einen grossen Handlungsspielraum offenlassen. Wie dieser Handlungsraum gefüllt wird, d. h. welche Entscheidungen konkret getroffen werden, ist, so die zweite These, wesentlich durch lokale Faktoren geprägt. Um auf unser Beispiel zurückzukommen: Die Entscheidung, eine Dezimal- anstatt wie damals üblich eine Dualmaschine zu bauen, war nicht eine anachronistische Fehlentscheidung, sondern zum grossen Teil durch die lokalen Besonderheiten bestimmt. Das gleiche gilt für eine Reihe weiterer Entscheidungen, die wir diskutiert haben.

Im Falle der ERMETH sind es vor allem zwei lokale Konstellationen, die eine wesentliche Rolle gespielt haben. Im Gegensatz zu anderen Maschinen, die in dieser Zeit gebaut wurden, war die Entwicklung der ERMETH spezifisch auf die *Mathematik* ausgerichtet. Die ERMETH war zuallererst ein mathematisches Projekt und verstand sich erst sekundär als ein Beitrag zur Computerentwicklung. Die Tatsache, dass die ERMETH im Kontext eines mathematischen Instituts konzipiert wurde und Mathematiker eine entscheidende Rolle spielten, hat sich auf viele Entscheidungen ausgewirkt und erklärt eine Reihe von konzeptionellen Besonderheiten, so z. B. die Gleitkomma-Arithmetik oder auch das Dezimalsystem. Ein zweiter spezifisch lokaler Faktor war die *Dienstleistungsfunktion*, die die ERMETH besass. Die spezifische Situation in der Schweiz, d. h. die Tatsache, dass es lange Zeit nur einen Computer gab und Aufträge mit Standardproblemen wegen der Kleinheit des Landes die Ausnahme waren, hat für das *Institut für angewandte Mathematik* die Gefahr mit sich gebracht, zu einem Dienstleistungszentrum zu werden und seine ursprüngliche Aufgabe nicht mehr erfüllen zu können. Die Externalisierung der Programmierarbeit war aus der Sicht von Stiefel die einzige Möglichkeit,

dies zu verhindern. Entsprechend wurde Benutzerfreundlichkeit zu einem zentralen Designprinzip, das viele Entscheidungen beeinflusste.

Dem ERMETH-Team ist es allerdings nicht gelungen, den lokalen Rahmen zu sprengen: Die gewonnenen Erkenntnisse wurden kaum generalisiert und diffundiert. Das Wissen blieb lokal und an konkrete Personen gebunden. Im Gegensatz etwa zur IAS-Maschine von John von Neumann, bei der eine gezielte Publikationspolitik verfolgt wurde mit dem Ziel, die Computerentwicklung in eine bestimmte Richtung zu lenken, blieb die ERMETH ein lokales Projekt, das gegen aussen hin wenig Ausstrahlung hatte. Als man, um ein Beispiel zu erwähnen, an der *Ecole d'ingénieur*, dem Vorläufer der ETH Lausanne, beschloss, sich Kompetenzen im Bereich des „automatischen Rechnens" zu beschaffen und eventuell selber eine Maschine zu kaufen, wurde der Emissär nicht etwa nach Zürich geschickt, sondern nach England zu Maurice Wilkes und nach Frankreich zu Louis Couffignal. Ein ähnliches Bild zeigt sich auch in Hinblick auf die Industrie. Obschon intendiert, gelang es nicht, die ERMETH kommerziell zu verwerten.

(3) *Handeln vs. Wissen.* Wie wir zu Beginn ausgeführt haben, dominiert in der Techniksoziologie eine wissenszentrierte Perspektive. Das ist umso erstaunlicher, als sich gerade im Bereich der Technik eine Analyse der konkreten Praktiken ganz besonders aufdrängen würde. Obschon die Beteiligten in ihren Beschreibungen der konkreten Entwicklungsarbeit vom Ideal des zielgerichteten und planvollen Handelns immer wieder auch abweichen, war es uns selbst aufgrund der Quellenlage nicht möglich, die skizzierte praxisorientierte Perspektive konsequent durchzuführen. Da ist von den verschiedenen „Tricklis" die Rede, die sie angewandt haben, vom Improvisieren und Experimentieren und von nachträglichen technischen Änderungen. D. h. die „kreative" Dimension des technischen (und konzeptionellen) Handelns scheint zwar in den Darstellungen deutlich auf, sie aber empirisch konsequent zu rekonstruieren, ist bei einer Studie, die auf historischem Material beruht, praktisch unmöglich. Um die „pragmatische Wende" auch für die Techniksoziologie zu realisieren, bedürfte es Untersuchungen, die – ähnlich wie es die Laborstudien für die Wissenschaftssoziologie getan haben – empirisch als Beobachtungstudien angelegt sind und theoretisch konsequent auf einer pragmatischen Handlungstheorie aufbauen.

LITERATURVERZEICHNIS

1. *Nicht veröffentlichte Quellen*

Automatisches Rechnen an der ETH, Zürich 1960.

Institut für Angewandte Mathematik (Hrsg.), Jahresberichte, 1949–1970.

Institut für angewandte Mathematik (1955), Liste der 1950–1955 mit der programmgesteuerten Rechenmaschine Z4 ausgeführten Aufträge und mathematischen Untersuchungen, Zürich.

Institut für angewandte Mathematik (1957), Orientierung über die elektronische Rechenmaschine der E.T.H. und deren Einsatz für technische Berechnungen, Zürich.

Memorandum über die Zweckbestimmung des Rechenzentrums der ETH, dazugehörige Randbedingungen und Gegebenheiten als Grundlage für die Planung des ORL-RZ-Neubaus am Zehnderweg, Zürich o. J.

Stiefel, Eduard (1949a), Gesuch an die Schweizerische Volkswirtschaftsstiftung, Zürich, 13. Oktober.

Stiefel, Eduard (1949b), Bericht über eine Studienreise nach den vereinigten Staaten von Amerika, 18. Oktober 1948 bis 12. März 1949, Zürich.

Stiefel, Eduard (1952), Bericht über ein Semester als Gastprofessor an der Universität von Californien und über die mathematischen Organisationen des „National Bureau of Standards", Zürich.

2. *Literatur*

Asdonk, Jupp; Udo Bredeweg und Uli Kowol (1991), Innovation als rekursiver Prozess, *Zeitschrift für Soziologie*, 20, 290–304.

Aspray, William (1986), International Diffusion of Computer Technology, 1945–1955, *Annals of the History of Computing*, 8/4, 351–360.

Aspray, William (1990a), *John von Neumann and the Origins of Modern Computing*, Cambridge, Mass.: MIT Press.

Aspray, William (Hrsg.) (1990b), *Computing Before Computers*, Ames: Iowa State University Press.

Baer, Jean-Loup (1980), *Computer Systems Architecture*, Rockville: Computer Science Press.

Bauer, Friedrich L. (1980), Between Zuse und Rutishauser. The Early Development of Digital Computing in Central Europe, in: Nicholas Metropolis, Jack Howlett und Gian-Carlo Rota (Hrsg.), *A History of Computing in the Twentieth Century*, New York: Academic Press, 505–524.

Belt, van den, Henk und Arie Rip (1987), The Nelson-Winter-Dosi Model and Synthetic Dye Chemistry, in: Wiebe E. Bijker, Thomas P. Hughes und Trevor Pinch (Hrsg.), *The Social Construction of Technological Systems*, Cambridge, Mass.: MIT Press, 135–158.

Bigelow, Julian (1980), Computer Development at the Institute for Advanced Study, in: Nicholas Metropolis, Jack Howlett und Gian-Carlo Rota (Hrsg.), *A History of Computing in the Twentieth Century*, New York: Academic Press, 291–310.

Bijker, Wiebe E. (1987), The Social Construction of Bakelite: Toward a Theory of Invention, in: Wiebe E. Bijker, Thomas P. Hughes und Trevor Pinch (Hrsg.), *The Social Construction of Technological Systems*, Cambridge, Mass.: MIT Press, 159–187.

Bijker, Wiebe E. (1992), The Social Construction of Fluorescent Lighting, or How an Artifact was Invented in Its Diffusion, in: Wiebe E. Bijker und John Law (Hrsg.), *Shaping Technology/Building Society. Studies in Sociotechnical Change*, Cambridge, Mass.: MIT Press, 75–102.

Bucciarelli, Louis L. (1994), *Designing Engineers*, Cambridge, Mass.: MIT Press.

Burks, Arthur W. und Alice R. Burks (1981), The ENIAC: First General-Purpose Electronic Computer, *Annals of the History of Computing*, 3/4, 310–389.

Burks, Arthur W.; Herman H. Goldstine und John von Neumann ([1946] 1987), Preliminary Discussion of the Logical Design of an Electronic Computing Instrument, in: William Aspray und Arthur W. Burks (Hrsg.), *Papers of John von Neuman on Computing and Computer Theory*, Cambridge Mass.: MIT Press, 97–142.

Callon, Michel (1983), Die Kreation einer Technik. Der Kampf um das Elektroauto, *Technik und Gesellschaft*, 2, Frankfurt/M.: Campus, 140–160.

Callon, Michel (1986), Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of St. Brieuc Bay, in: John Law (Hrsg.), *Power, Action, and Belief: A New Sociology of Knowledge?* London: Routledge and Kegan Paul, 196–233.

Ceruzzi, Paul E. (1983), *Reckoners: The Prehistory of Digital Computer. From Relays to the Stored Program Concept, 1935–1945,* Westport, Conn.: Greenwood Press.

Ceruzzi, Paul E. (1986), An Unforeseen Revolution: Computers and Expectations, 1935–1985, in: Joseph J. Corn (Hrsg.), *Imagining Tomorrow*, Cambridge, Mass.: MIT Press, 188–201.

Ceruzzi, Paul E. (1991), When Computers Were Human, *Annals of the History of Computing*, 13/3, 237–244.

Cohen, Bernard I. (1987), Howard H. Aiken and the Computer, in: Stephen C. Nash (Hrsg.), *A History of Scientific Computing*, New York: ACM Press, 41–52.

Cohen, Bernhard I. (1988), The Computer: A Case Study of Support by Government, Especially the Military, of a New Science and Technology, in: Everett Mendelsohn, Merritt Roe Smith und Peter Weingart (Hrsg.), *Science, Technology and the Military*, Dordrecht: Kluwer, 119154.

Collins, H.M. (1981), Stages in the Empirical Programme of Relativism, in: Ders. (Hrsg.), *Knowledge and Controversy: Studies of Modern Natural Science*, Social Studies of Science, 11, 1, 3–10.

Comrie, L.J. (1944), Careers for Girls, *The Mathematical Gazette*, 90–95.

Constant, Edward W. (1973), A Model of Technological Change Applied to the Turbojet Revolution, *Technology and Culture*, 14, 553–572.

Coolidge, C.A. (1953), *Design of an Automatic Digital Calculator*, Cambridge, Mass.: Harvard University.

Dasgupta, Subrato (1984), *The Design and Description of Computer Architectures*, New York: John Wiley.

Dierkes, Meinolf; Ute Hoffmann und Lutz Marz (1992), *Leitbild und Technik. Zur Entstehung und Steuerung technischer Innovationen*, Berlin: Sigma.

Dierkes, Meinolf; Weert Canzler, Lutz Marz und Andreas Knie (1995), Politik und Technikgenese, *Mitteilungen*, Verbund Sozialwissenschaftliche Technikforschung, 15, 7–28.

Dosi, Giovanni (1982), Technological Paradigms and Technological Trajectories, *Research Policy*, 11, 147–162.

Dosi, Giovanni (1988), The Nature of the Innovative Process, in: Giovanni Dosi u. a. (Hrsg.), *Technical Change and Economic History*, London and New York: Pinter, 221–238.

Engeler, Ernst u. a. (1981), *Konrad Zuse und die Frühzeit des wissenschaftlichen Rechnens an der ETH Zürich.* Dokumentation zu einer Ausstellung um die Z4, Mathematisches Seminar, ETH Zürich.

Flynn, Michael J. (1972), Some Computer Organization and Their Effectiveness, *IEEE Transactions on Computers*, C–21/9, 948–960.

Fujimura, Joan (1992), Crafting Science: Standardized Packages, Boundary Objects, and „Translation", in: Andrew Pickering (Hrsg.), *Science as Practice and Culture*, Chicago: University of Chicago Press, 168–211.

Furger, Franco (1992), *Informatik-Innovationen aus der Schweiz? Lilith/Diser und Oberon*, Zürich: Verlagsgemeinschaft Technopark.

Galison, Peter (1988), History, Philosophy, and the Central Metaphor, *Science in Context*, 2, 197–212.

Giloi, Wolfgang K. (1981), *Rechnerarchitektur,* Berlin u. a.: Springer.

Goldstine, Herman H. (1972), *The Computer from Pascal to von Neumann*, Princeton: Princeton University Press.

Gooding, David (1992), Putting Agency Back into Experiment, in: Andrew Pickering (Hrsg.), *Science as Practice and Culture*, Chicago: Chicago University Press, 65–112.

Gutknecht, Martin H. (1987), The Pioneer Days of Scientific Computing in Switzerland, *ACM Conference on the History of Scientific and Numeric Computation.* Conference Proceedings, New York, 63–69.

Gutting, Gary (1984), Paradigms, Revolutions, and Technology, in: Rachel Laudan (Hrsg.), *The Nature of Technological Knowledge: Are Models of Scientific Change Relevant?* Dordrecht: Reidel, 47–65.

Hard, Mikael (1994), Technology as Practice: Local and Global Closure Processes in Diesel-Engine Design, *Social Studies of Science,* 24, 549–585.

Heims, Steve J. (1991), *The Cybernetics Group*, Cambrige, Mass.: MIT Press.

Heintz, Bettina (1993a), Wissenschaft im Kontext. Neuere Entwicklungstendenzen in der Wissenschaftssoziologie, *Kölner Zeitschrift für Soziologie und Sozialpsychologie*, 45/3, 528–552.

Heintz, Bettina (1993b), *Die Herrschaft der Regel. Zur Grundlagengeschichte des Computers*, Frankfurt/M.: Campus.

Heintz, Bettina (1997), Die soziale Welt der Wissenschaft. Entwicklungen, Ansätze und Ergebnisse der Wissenschaftsforschung, in: Bettina Heintz und Bernhard Nievergelt (Hrsg.), *Wissenschafts- und Technikforschung in der Schweiz*, Zürich: Seismo.

Heintz, Bettina (1998), *Die Innenwelt der Mathematik. Epistemische Praktiken und kultureller Kontext*, Berlin u. a.: Springer (im Erscheinen).

Hug, Peter (1997), Atomtechnologieentwicklung in der Schweiz zwischen militärischen Interessen und privatwirtschaftlicher Skepsis, in: Bettina Heintz und Bernhard Nievergelt, *Wissenschafts- und Technikforschung in der Schweiz*, Zürich: Seismo.

Kidder, Tracy (1984), *Die Seele einer Maschine. Vom Entstehen eines Computers*, Reinbek b. Hamburg: Rowohlt.

Kowol, Uli und Wolfgang Krohn (1995), Innovationsnetzwerke. Ein Modell der Technikgenese, *Technik und Gesellschaft*, 8, Frankfurt/M.: Campus, 77–106.

Knorr-Cetina, Karin (1984), *Die Fabrikation von Erkenntnis*, Frankfurt/M.: Suhrkamp.

Knuth, Donald E. und Luis Trabb Pardo (1980), The Early Development of Programming Languages, in: Nicholas Metropolis; Jack Howlett und Gian-Carlo Rota (Hrsg.), *A History of Computing in the Twentieth Century*, New York: Academic Press, 197–274.

Latour, Bruno (1987), *Science in Action*, Cambridge: Harvard University Press.

Latour, Bruno (1988a), *The Pasteurization of France*, Cambridge, Mass.: Harvard University Press.

Latour, Bruno (1988b) Drawing Things Together, in: Michael Lynch und Steve Woolgar (Hrsg.), *Representation in Scientific Practice*, Cambridge, Mass.: MIT Press, 19–68.

Läuchli, Peter (1993), Erinnerungen an die Pionierzeit der Informatik, ETH Zürich, *Kleine Schriften*, Nr. 23.

Laudan, Rachel (Hrsg.) (1984), *The Nature of Technological Knowledge: Are Models of Scientific Change Relevant?* Dordrecht: Reidel.

Layton, Edwin T. (1974), Technology as Knowledge, *Technology and Culture*, 15/1, 31–41.

Lavington, Simon Hugh (1980), *Early British Computers,* Manchester: Manchester University Press.

Lester, B.R. (1951), General Electric Engineering Digital Computer. Proceedings of a Second Symposium on Large-Scale Digital Computing Machinery, *Annals of the Computation Laboratory of Harvard University*, 26, Cambridge, Mass.

MacKenzie, Donald (1989), From Kwajalein to Armageddon? Testing and the Construction of Missile Accuracy, in: David Gooding; Trevor Pinch und Simon Schaffer (Hrsg.), *The Uses of Experiment: Studies in the Natural Sciences*, Cambridge: Cambridge University Press, 409–435.

Mannheim, Karl ([1931] 1969), Wissenssoziologie, in: Ders., *Ideologie und Utopie*, Frankfurt/M.: Vittorio Klostermann, 227–267.

McCulloch, Warren S. und Walter H. Pitts ([1943] 1990), A Logical Calculus of the Ideas Immanent in Nervous Activity, in: Margaret A. Boden (Hrsg.), *The Philosophy of Artificial Intelligence*, Oxford: University Press, 22–39.

Metropolis, Nicholas; Jack Howlett und Gian-Carlo Rota (Hrsg.) (1980), *A History of Computing in the Twentieth Century*, New York: Academic Press.

Naur, Peter (1981), The European Side of the Last Phase of the Development of ALGOL, in: Richard L. Wexelblat (Hrsg.), *A History of Programming Languages*, New York: Academic Press, 92–139.

Neumann, John von ([1945] 1987), First Draft of a Report on the EDVAC, in: William Aspray und Arthur W. Burks (Hrsg.), *Papers of John von Neuman on Computing and Computer Theory,* Cambridge Mass.: MIT Press, 17–82.

Neumann, John von ([1946] 1981), The Principles of Large-Scale Computing Machines, *Annals of the History of Computing*, 3/3, 263–273

Newell, Allen (1986), The Symbol Level and the Knowledge Level, in: Zenon W. Pylyshyn und William Demopoulos (Hrsg.), *Meaning and Cognitive Structure: Issues in the Computational Theory of Mind*, Norwood: Ablex, 31–39.

Olazaran, Mikel (1996), A Sociological Study of the Official History of the Perceptrons Controversy, *Social Studies of Science*, 26, 611–659.

Petzold, Hartmut (1985), *Rechnende Maschinen. Eine historische Untersuchung ihrer Herstellung und Anwendung vom Kaiserreich bis zur Bundesrepublik*, Düsseldorf: VDI-Verlag.

Petzold, Hartmut (1987), Die Entstehung der elektronischen Industrie in Deutschland und den USA. Der Beginn der Massenproduktion von Elektronenröhren 1912–1918, *Geschichte und Gesellschaft*, 13, 340–367.

Pickering, Andrew (1992), From Science as Knowledge to Science as Practice, in: Andrew Pickering (Hrsg.), *Science as Practice and Culture*, Chicago: Chicago University Press, 1–26.

Piloty, H. und R. Piloty (1953), *Leitgedanken, Ziel und Stand der Müchner Entwicklung.* Vorträge über Rechenanlagen, Max-Planck-Institut für Physik, Göttingen.

Pinch, Trevor J. und Wiebe E. Bijker (1987), The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other, in: Wiebe E. Bijker u. a. (Hrsg.), *The Social Construction of Technological Systems. New Directions in the Sociology and History of Technology*, Cambridge, Mass.: MIT Press, 17–50.

Pugh, Emerson W. (1995), *Building IBM: Shaping an Industry and Its Technology*, Cambridge, Mass.: MIT Press.

O'Connell, Joseph (1993), Metrology: The Creation of Universality by the Circulation of Particulars, *Social Studies of Science*, 23, 129–173.

Rammert, Werner (1995), Regeln der technikgenetischen Methode. Die soziale Konstruktion der Technik und ihre evolutionäre Dynamik, *Technik und Gesellschaft*, 8, Frankfurt/M.: Campus, 13–30.

Ramunni, Jérôme (1989), *La physique du calcul*, Paris: Hachette.

Randell, Brian (Hrsg.) (1982), *The Origins of Digital Computers*, Berlin u. a.: Springer.

Rosenberg, Nathan (1983), *Inside the Black Box: Technology and Economics*, Cambridge: Cambridge University Press.

Rutishauser, Heinz (1950), Die neueste elektronische Rechenmaschine, in: *Neue Zürcher Zeitung*, 26. April, Nr. 871.

Rutishauser, Heinz (1951), Über automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen, in: *ZAMM*, 31, 254–255.

Rutishauser, Heinz (1952), *Automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen*, Mitteilungen Nr. 3 aus dem Institut für angewandte Mathematik der ETH Zürich, Basel: Birkhäuser.

Rutishauser, Heinz; Ambros Speiser und Eduard Stiefel (1951), *Programmgesteuerte digitale Rechengeräte (elektronische Rechenmaschinen)*, Sonderheft der Zeitschrift für angewandte Mathematik und Physik, 2, Basel: Birkhäuser.

Schai, Alfred (1957), Die elektronischen und magnetischen Schaltungen der ERMETH, *Scientia Electrica*, 3, 4, 127–140.

Schwarz, H.R. (1981), The Early Years of Computing in Switzerland, *Annals of the History of Computing*, 3/2, 121–132.

Slotnick, D.L. (1982), The Conception and Development of Parallel Processors – A Personal Memoir, *Annals of the History of Computing*, 4/1, 20–30.

Smith, Richard E. (1989), A Historical Overview of Computer Architecture, *Annals of the History of Computing*, 10/4, 277–303.

Smith, Thomas M. (1976), Project Whirlwind: An Unorthodox Development Project, *Technology and Culture*, 17/3, 447–464.

Speiser, Ambros (1950), Das programmgesteuerte Rechengerät an der Eidgenössischen Technischen Hochschule in Zürich, *Neue Zürcher Zeitung*, 30. August, Nr. 1796.

Speiser, Ambros (1979), Digital Processing: Reflections About the Past and the Future, *Journal of Applied Mathematics and Physics*, 30, 355–363.

Speiser, Ambros, die Zukunft der Computer aus der Sicht des Jahres 1950. Rückblick auf ein Stück „Technological Forecasting", *Neue Zürcher Zeitung*, Juni 1985, Nr. 127.

Speiser, Ambros (1992), 95 Semester ETH. Der Weg zur Informatik, ETH Zürich, *Kleine Schriften*, Nr. 22.

Star, Susan Leigh und James Griesemer (1989), Institutional Ecology, „Translations", and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–1993, *Social Studies of Science*, 19, 387–420.

Stern, Nancy (1981), *From ENIAC to UNIVAC*, Bedford: Digital Press.

Stiefel, Eduard (1948), Elektronische Rechenmaschinen, *Neue Zürcher Zeitung*, 13. Oktober, Nr. 2140.

Stiefel, Eduard (1953), La machine à calculer arithmétique „Z4" de l'Ecole Polytechnique Fédérale à Zürich (Suisse) et son application à la résolution d'une équation aus dérivées partielles de type elliptique, in: Colloque International du CNRS, XXXVII, *Les machines à calculer et la pensée humaine*, Paris, 8–13 janvier 1951, herausgegeben vom C.N.R.S., Paris, 33–40.

Stiefel, Eduard (1956), Rechenautomaten im Dienste der Technik, *Arbeitsgemeinschaft für Forschung des Landes Nordrhein-Westfalen*, Nr. 45, 29–46.

Stiefel, Eduard (1957), Einsatz der Rechenautomaten in der Technik, in: *Bulletin des Schweizerischen Elektrotechnischen Vereins*, 48/11, 507– 513.

Stock, John Robert (1956), *Die mathematischen Grundlagen für die Organisation der elektronischen Rechenmaschine der Eidgenössischen Technischen Hochschule*, Mitteilungen des Instituts für Angewandte Mathematik, Nr. 6, Basel–Stuttgart: Birkhäuser.

Suchman, Lucy A. (1987), *Plans and Situated Actions. The Problem of Human-Machine Communication*, Cambridge: Cambridge University Press.

Vincenti, Walter G. (1995), The Technical Shaping of Technology: Real-World Constraints and Technical Logic in Edison's Electrical Lighting System, *Social Studies of Science*, 25, 553–574.

Wagner, Gerald (1994), Vertrauen in Technik, *Zeitschrift für Soziologie*, 23/2, 145–157.

Weingart, Peter (1982), Strukturen technologischen Wandels. Zu einer soziologischen Analyse der Technik, in: Rodrigo Jokisch (Hrsg.), *Techniksoziologie*, Frankfurt/M.: Suhrkamp, 112 – 141.

Wilkes, Maurice (1985), *Memoirs of a Computer Pioneer*, Cambridge, Mass.: MIT Press.

Wilkes, Maurice (1956), *Automatic Digital Computers*, London: Methuen.

Williams, Michael Roy (1985), *A History of Computing Technology*, Englewood Cliffs NJ.: Prentice-Hall.

Winograd, Terry, Fernando Flores (1986), *Understandig Computers and Cognition*, Reading, Mass.: Addison Wesley.

Zuse, Konrad (1984), *Der Computer – Mein Lebenswerk*, Berlin u. a.: Springer.

Zuse, Konrad (1992), *Computerarchitektur aus damaliger und heutiger Sicht*, Zürich: Abteilung für Informatik, ETH Zürich.

Anschrift der Autoren:
Dr. Franco Furger
Institute of Public Policy
George Mason University
4400 University Drive
Fairfax, VA 22030-4444, USA

Prof. Dr. Bettina Heintz
Institut für Soziologie
Johannes Gutenberg-Universität Mainz
Colonel-Kleinmann-Weg 2
D-55099 Mainz