

Computeralphabetismus

Friedrich Kittler

Alles spricht heute dafür, daß Wolken, Meere und alle anderen Dinge halbwegs zwischen dem Schönen und dem Erhabenen nicht berechenbar sind – und zwar im strikten, computertheoretischen Wortsinn von Berechenbarkeit. Also kann nichts und niemand, auch kein Computer, irgendeine Zukunft prophezeien. Möglich sind nur Warnungen, die den Stand des Schreibens über das Schreiben – also die Literatur – am Stand der technischen Dinge messen.

Auch das Computerzeitalter in jenen heroischen Gründerjahren, als noch kein Verkaufserfolg seine Fremdheit trübte und binärer Code ein Fremdwort war, begann als Literatur. Turings Dissertation beschrieb 1936 das Prinzip aller Computer im Begriff einer Papiermaschine, der genauso gut Dissertationen einschloß. Zwei Jahre später legte ein junger Student namens Claude Elwood Shannon eine Magisterarbeit vor, die als »folgenreichste Magisterarbeit aller Zeiten (vgl. Hagemeyer 1979: 432) zum Beweis antrat, daß sämtliche logischen Funktionen, auf die die Menschengattung so stolz ist, von schlichten Relaischaltern eines Telegraphen- oder Telephonnetzes übernommen werden können (vgl. Shannon 1938). Damit aber sind die Papiermaschinen ihrem eigenen Medium entsprungen. Shannons Beweis hat nicht nur den Ausgang des Zweiten Weltkriegs und die technische Konstruktion von Computern mitbestimmt, sondern auch die nutzloseste Maschine aller Zeiten hervorgebracht. Zur selben Zeit nämlich, als er die Magisterarbeit schrieb, baute Shannon in seiner Studentenbude einen schlichten schwarzen Kasten, an dem nur ein einziger Schalter mit der Aufschrift ON/OFF zu sehen war. Wenn Freunde zu Besuch kamen, konnten sie der Versuchung kaum widerstehen, diesen

Schalter, der immer auf Aus stand, in die Stellung Ein zu versetzen. Also gleich hob sich der Deckel des schwarzen Kastens, eine mechanische Hand tauchte aus der Tiefe auf, schwenkte langsam nach unten, tastete die Seitenwand des Kastens ab, fand den Schalter und stellte ihn wieder auf Aus. Woraufhin das ganze Spiel in umgekehrter Richtung ablief: Die Hand kroch zurück nach oben, tauchte in ihren Kasten und zog den Deckel hinter ihr zu; alles war wie zu Anfang und konnte von neuem beginnen ... Und wirklich: Shannons schlechte Unendlichkeit, wie Hegel gesagt hätte, startete mit ihrer endlosen Abfolge von Ein und Aus, Ja und Nein, Eins und Null das neue, digitale Zeitalter. Das nutzloseste Spielzeug auf Erden hat Geschichte gemacht. Sicher gab es auch vorher binäre Elemente, an Geräten wie der Schreibmaschine oder an Kleidungsstücken wie dem Netzstrumpf (vgl. Hornung 1989: 258), aber noch nie war eine ganze Maschine darin aufgegangen, Schalter zu sein. Irgendwo hatten alle Maschinen vor Shannon, womöglich weil sie einer Körpermotorik treu blieben, Teile mit stetiger Bewegung gehabt. Und falls auch diskrete Elemente vom Schaltertyp eingebaut waren, stammten sie vermutlich von jener fundamentalen Artikulation ab, mit der die Vokalalphabeten als Europas wahrhafter Ursprung das große Reich akustischen Rauschens einst gegliedert haben. Noch bei Shannons Spielzeug vertraten ja die zwei Aufschriften ON und OFF eine binäre Schaltlogik nach außen, nämlich gegenüber möglichen Benutzern, die ersichtlich Leser waren.

Aber daß eine Maschine schlechthin nichts anderes tat, als zwischen ihren zwei Schaltzuständen hin- und herzuwechseln, hatte es noch nie gegeben. Die Finite-State-Maschinen namens PC, die heute auf jedem zweiten Schreibtisch stehen, gehen auf eine Revolution zurück. Der große Mathematiker Norbert Wiener, obwohl oder weil er beim Aufbau der modernen Informatik in enger Zusammenarbeit mit Shannon stand, soll über seinen Kollegen gesagt haben: »Shannon ist einfach verrückt, er denkt nur algebraisch.«¹ Während Wiener, in bester mathematischer Tradition, endlose Differentialgleichungen aufstellen konnte, die er seinen Auftraggebern im Pentagon, schneller als sie mitschreiben konnten, direkt über Telefon durchgab, reduzierte der verrückte Shannon auch und gerade die Mathematik aufs kleinste aller denkbaren Alphabete. (Weshalb die von Shannon begründete Kommunikationstheorie nur diskrete Nach-

¹ Vgl. die höflicheren Formulierungen bei Wiener (1962: 155 und 228). Klassisches Beispiel dieses Wahnsinns ist und bleibt die Gleichung »1+1=1« (Shannon 1938: 713)

richtensysteme berechnen kann und den analogen Fall vom diskreten extrapolieren muß.) Insofern ist das Computerzeitalter, zumindest bei seinen Stiftern und Programmierern, die Vollendung des europäischen Alphabetismus. Die zwei wichtigsten Steuersignale, die eine Zentrale Recheneinheit mit ihrem externen Speicher verbinden, heißen üblicherweise LESEN und SCHREIBEN oder genauerhin READ und WRITE. Man muß also, wie es so schön heißt, des Englischen mächtig sein, um Datenblätter lesen und das heißt eine Maschine entziffern zu können, die ihrerseits wiederum schreibt und liest. Nicht umsonst hatte Turing, der englische Mathematiker und Kryptoanalytiker, das Prinzip aller möglichen Computer zunächst als Papiermaschine angeschrieben. Diese Papiermaschine nämlich tut genau das, was Mathematiker tun, wenn sie Zahlen und Buchstaben auf einem Stück Papier manipulieren, nur eben von allein. Nicht umsonst auch soll Turings Handschrift als Schüler so grauenhaft gewesen sein, daß sie ihm sämtliche Mathematikensuren verdarb und statt dessen, in schöner Folgerichtigkeit, die Konstruktion einer »ungemein primitiven« Schreibmaschine eingab (vgl. Hodges 1983: 14). An die Stelle dieser nie gebauten Schreibmaschine trat dann im Zweiten Weltkrieg Turings Prototyp-Computer mit einem ebenso kriegsentscheidenden wie geglückten Auftrag, die verschlüsselten UKW-Funksprüche der deutschen Wehrmacht in Echtzeit zu knacken. Schreibmaschinen, mit anderen Worten, haben nur das Schreiben mechanisiert, Computer sparen darüber hinaus auch noch die Sekretärin als Leserin ein.

Deshalb stammen alle heutigen Programmierer von Turing und Shannon, den Codebrechern des Zweiten Weltkriegs, ab. Aus Codebreakers, wie es in ihrer einzigen Weltgeschichte heißt (vgl. Kahn 1967), sind Codemakers geworden, Programmierer also, die keine Texte mehr schreiben, sondern Source Listings. Und das ganze Geheimnis solcher Quellcodes läuft darauf hinaus, Maschinen mit einer Beschreibung anderer Maschinen zu füttern, die allesamt Shannons endloses ON/OFF repetieren. Als Quellcode implodiert der Alphabetismus unserer Kultur zu einer Schrift, die alle Züge eines Geheimcodes trägt, weil nicht Leute, sondern Computer sie lesen können müssen. Dieses noch nie dagewesene Schreiben läßt sich am einfachsten mit einem Zitat von Karl-Herrmann Rollke erläutern. Besagter Rollke hat als Gymnasiallehrer für Mathematik eines jener seltenen deutschen Programmierhandbücher geschrieben, die ohne Leerlauf und Wiederholung auskommen. Wenn Rollkes TURBO PASCAL 5.0 BUCH auf das klassische Thema von Dateien und Listen zu sprechen

kommt, heißt es zur Erläuterung eines exemplarischen Stücks Quellcodes ganz lapidar: »Als nächstes« – nämlich nach einer Prozedur namens Lesen – »schreiben wir die Prozedur Schreiben« (Rollke 1989: 216). Diese nur scheinbare Tautologie demonstriert schlagartig, was das Computerzeitalter, gerade in bezug auf Alphabetisierung, von der gesamten europäischen Schriftkultur unterscheidet. Man schreibt nicht mehr, man bringt auch keinen Schülern mehr das Schreiben bei. Die Schrift bleibt vielmehr einem Automaten überantwortet, dem man unter Titeln wie Prozedur, Funktion oder Routine eine Beschreibung seines Schreibens geschrieben hat. »Auch die Prozedur Schreiben«, heißt es bei Rollke weiter, »hat zwei Parameter und macht nichts anderes, als einen Datensatz mit der Nummer N und dem Inhalt SATZ auf die Diskette zu schreiben.« (Rollke 1989: 216)

Nur hat gerade diese Einfachheit auf seiten der Maschine als Kehrseite, auf seiten des sogenannten Menschen nämlich, die Tücke, lauter Kompliziertheiten, Ungewohntheiten und Fehlerträchtigkeiten hervorzurufen. Man schreibt die Prozedur Schreiben nicht schlichtweg hin wie den berühmten Satz, der laut Paul Valéry eigentlich alle Romane eröffnen könnte: »Die Marquise verließ das Haus um 5 Uhr.« Die Prozedur Schreiben, um überhaupt den Standards einer Programmiersprache wie Turbo Pascal zu entsprechen, sollte vielmehr als Prozedur deklariert sein; ihre sogenannten Parameter, die besser Argumente heißen, sollten alle und zudem noch korrekt einem in Größe und Struktur wohldefinierten Datentyp zugeordnet sein; schließlich und endlich muß die Prozedur, auf Gedeih und Verderb, mit den drei Buchstaben END und einem folgenden Semikolon abgeschlossen werden. Wohlgemerkt mit Semikolon, weil Kommas in Turbo Pascal vollkommen andere Funktionen haben. Ein Satzzeichen also, das in amerikanischen Büchern fast ausgestorben ist und in deutschen gerade noch einen pädagogisch gehegten Untod führt, feiert Urstände, und zwar keine fröhlichen, sondern ausgesprochen strikte. Wer jenes Semikolon vergessen oder mit dem Komma verwechseln würde, käme im gemeinen Leben ungestraft und in der gemeinen Schule bestenfalls mit einem roten Strich am Heftrand davon. In Programmiersprachen dagegen ist ein einziges falsches Zeichen durchaus imstande, Abstürze hervorzurufen: Im harmlosesten Fall streikt die maschinelle Übersetzung, weil sie den Fehler erkennt, aber nicht beheben und folglich auch kein lauffähiges Programm erstellen kann. In schlimmeren Fällen, wo der Computer als Leser seiner eigenen Steuerbefehle keinen Fehler entdecken konnte, kann das laufende Programm abstürzen und im schlimmsten Fall

(oder worst case), wenn nämlich auch das Betriebssystem keinen Fehler erkennt, die Maschine im ganzen. »Von einem Wort«, wußte schon Goethes Mephisto, »läßt sich kein Iota rauben.« Diese neue Scholastik der Computercodes fördert im Effekt also den ältesten Buchstabengehorsam wieder ein. Während moderne Menschen (nach Nietzsches bösem Wort) unter dem, was geschrieben steht, so ziemlich alles außer eben jenem Wortlaut verstehen dürfen (vgl. Nietzsche 1967 : 115 f.), nur damit Schulen, Universitäten und Kongresse nicht arbeitslos werden, läuft jeder Lehrgang für Programmierer darauf hinaus, absolut willkürliche Zuordnungen von Buchstaben zu Zahlen auswendig zu lernen, nur damit das System nicht abstürzt. Digitale Maschinen zahlen für ihre Unfehlbarkeit, was die vereinbarte Rechengenauigkeit angeht, also mit einer denkbar hohen Fehlerträchtigkeit beim Schreiben und Lesen von Programmen. Wer statt Romanen oder Vorträgen Quellcode verfaßt, verschenkt üblicherweise drei Viertel seiner Zeit mit Debuggen oder Wanzenjagen, wie die Fehlersuche an Computern seit jenen mythischen Tagen heißt, als noch wirkliche Wanzen die Elektronenröhren eines Pentagon-Dinosauriers lahmgelegt haben sollen. Und wer kommerzielle Software kauft, muß von vornherein damit rechnen, nach einer Laufzeit zwischen dreißig und dreihundert Stunden dem ersten Programmierfehler zu begegnen. Dringlicher als das Schreiben von Programmen sind Entwanzen und Warten geworden. An ihnen führt kein Weg vorbei, auch nicht der neuerliche Versuch, fehlertolerante Systeme zu entwickeln, weil die Fehlertoleranz selber ja nicht fehlertolerant eingegeben werden kann.

In dieser Notlage, nicht für Leser, sondern für Computer zu schreiben, bleibt Programmierern nur Computeranalphabetismus übrig. So soll die seltsame Fähigkeit heißen, alles das, was gemeinhin als Denken und Rechnen läuft, auf Fähigkeiten und Unfähigkeiten einer digitalen Maschine abzubilden. Der Stil ist der Mensch selber, hatte einst die Philosophie der Aufklärung triumphal verkündet; der Stil ist immer nur der Mensch, den man adressiert, stellte die moderne Psychoanalyse richtig (vgl. Lacan 1966: 9); den Stil von Computerprogrammen bestimmen folglich keine Menschen mehr, sondern die Maschinen selbst. Nicht zum ersten Mal in der Medienweltgeschichte, die ja schon Telegrammstil und Zeitungsdeutsch hervorgebracht hat, aber in noch nie gesehener Strenge, ist Geschriebenes eine Funktion des Schreibzeugs. In Fesseln tanzen, hätte Nietzsche gesagt. Die Fesseln, um ein paar Beispiele anzuführen, reichen von der schlichsten Ebene bis zur grundsätzlichen. Eine ebenso elemen-

tare wie unerwähnte Begrenzung von Quellcodes liegt etwa darin, daß neben dem Semikolon auch die seltsamsten Schreibmaschinensonderzeichen, vom Dollarzeichen bis zum Klammeraffen, wiederauferstehen, aber kein einziger Umlaut akzeptiert wird. Mit anderen Worten: die amerikanische Tastatur, ein ASCII-Code von gerade 128 Zeichen, hat als weltweiter Standard triumphiert. Sie reicht bis in die Schreibweise mathematischer Gleichungen, die ihre ganze Eleganz oder Übersichtlichkeit einbüßen, einfach weil es auf zeilenorientierten Benutzeroberflächen kaum Möglichkeiten gibt, Zeichen übereinander oder untereinander einzugeben. Wie bei Shannons digitalem Spielzeug kann an derselben Stelle ja immer nur ein Zustand, immer nur ein Zeichen bestehen, was seinen Vorgänger automatisch ausradiert.

Genau diese Logik wandert als Computeralphabetismus aus Maschinen in Köpfe. In der Blütezeit der neuzeitlichen Mathematik war es ihr ganzer Stolz, auf dem Schreibpapier noch viel kühnere Dinge als Dichter und Maler zusammen anzustellen. Wie Vilém Flusser immer wieder betont hat, war ihr alphanumerischer Code, ein Code also gleichzeitig aus Zahlen und Buchstaben, dem bloßen Buchstabenschatz der Schriftsteller an Mächtigkeit, und das heißt an Spielmöglichkeiten, überlegen. Ein einziges mathematisches Kürzel, etwa der Buchstabe π , stand mühelos für eine Folge unendlich vieler Ziffern, wie nichts und niemand sie hätte anschreiben können. Idealität in der Neuzeit, im radikalen Unterschied zu den ganzen Zahlen des platonischen Ideals, hauste im Begriff der reellen Zahl (vgl. Spengler 1923: 100 - 105). Digitalcomputer dagegen sind beliebig, aber nie unendlich genau. Sie verkraften nur berechenbare Zahlen und lösen nur Aufgaben mit angebbaren vielen Schritten, also genau das, was seit ihrer Erfindung Algorithmus heißt. Weshalb Computeralphabeten die seltsame Fähigkeit entwickeln müssen, diese endlich vielen Schritte einen nach dem anderen zu machen. Seit Shannons kalter Ekstase helfen Erleuchtungen und Einfälle nicht mehr weiter, weil gerade umgekehrt die vollendete Dummheit nottut. Programmschleifen, obwohl und weil sie elegant sind, aufrollen, feindliche Divisionen, weil sie Rechenzeit verschlingen, durch schnellere Multiplikationen ersetzen, geschlossene Ausdrücke in numerische überführen: Das sind die dummen, aber effektiven Optimierungsstrategien, die aus Computerprogrammen und ihren Schreibern schließlich monotone lineare Befehlsfolgen machen. John von Neumanns Befehl, Daten und Befehle von Computerspeichern einfach aneinanderzureihen, hat das Denken nicht verschont.

»Sehen wir«, befiehlt deshalb ein weitverbreitetes Programmierhandbuch, »sehen wir den Tatsachen ins Auge: man kann schließlich kein ganzes Programme auf einmal im Kopf behalten (außer es handelt sich um ein ungewöhnlich kleines)« (Borland International 1989: 14). Mit dem, was Philosophen einst als Anschauung oder Intuition gefeiert haben, geht es also beim Programmieren zu Ende, allerdings nicht ohne das hübsche Paradox, daß Menschen und näherhin Programmierer dieser Tatsache ins Auge blicken sollen, was schließlich auch auf Anschauung hinausläuft. Aber just diese Wiederkehr, diese Rekursion des Problems im Augenblick seiner Formulierung führt schlagend vor, welche Entwöhnung Computeranalphabetismus verlangt und wie mühsam mithin seine Einübung ist. Vor Zeiten, nach der Durchsetzung des Buchdrucks als technischer Alphabetisierung Europas, wandten Pädagogen wie Comenius ihre ganze Liebe darauf, das gedruckte und deshalb in ihren Augen tote Buchwissen wieder lebendig zu machen. Zu diesem Zweck koppelten sie die Buchstaben, etwa im ORBIS PICTUS, bekanntlich mit einer optischen Anschauung, die hinterm Rücken jener pädagogischen Menschenliebe freilich auch nichts anderes als Drucksache war. Heute kommt es umgekehrt darauf an, den Sachverhalt hinzunehmen, daß Codes existieren, die keinen Menschen und erst recht nicht dessen Anschauung adressieren. So endet der Traum, alle Nachrichten hätten von einem selbst gesendet oder (nach den Symmetriannahmen gängiger Kommunikationstheorien) empfangen werden können. Was Philosophen seit Locke als Kommunikation oder Konsens feiern, ist wohl nur letzter vergeblicher Widerstand gegen die radikale Fremdheit der Zeichen, mit denen die nächsten Jahrhunderte werden auskommen müssen. Es gibt Codes; ob sie lesbar und schreibbar sind, steht auf einem anderen Blatt.

In einem Roman Thomas Pynchons verbringt ein deutscher Ingenieur, Kurt Mondaugen mit Namen, seine südwestafrikanischen Nächte kurz nach dem Ersten Weltkrieg damit, im Auftrag der TU Dresden das Hochfrequenzrauschen des Weltalls per Radio aufzufangen und zu analysieren. Nur gelingt es ihm nicht, irgendeine Ordnung, irgendeinen Code zu entdecken. Bis ein zweifelhafter Bekannter vorbeikommt, die endlosen mitgeschriebenen Buchstabenkolonnen noch einmal durchgeht und im Klartext anschreiben kann: »DIEWELTISTALLESWASDERFALL IST« (vgl. Pynchon 1981: 258). Wittgensteins berühmter Satz kommt mithin als Signal aus dem Äther, was aber nicht den tröstlichen Zuspruch gibt, der Kosmos selber philosophiere, sondern den Zufall aller Codierung nur

noch einmal unterstreicht. Diese historische Lage ist nicht schlechthin neu, denn nur die Anstrengung aller modernen Schulsysteme, vergessen zu machen, daß Lesen und Schreiben Anstrengungen und keine Naturgaben sind, sorgt für die Illusion, es gäbe ein natürliches Zeichensystem. Aber weil Computercodes aus den beschriebenen Gründen solche Illusionen gar nicht erst aufkommen lassen, trennen sie, schneidender als alle Buchstabenkulturen, zwischen einer alphanumerischen Elite und dem Rest der Welt. Der Code wird, ganz wie in jenen vorgeblich längst überwundenen Zeiten, als Kleriker den Buchstabenkundigen und Laien einfach als Analphabeten bezeichneten, zur Demarkationslinie. Sogar längst vergessene Unterscheidungen wie die zwischen Schreib- und Lesefähigkeit kehren mit einem Mal wieder. Als ein berühmter Ritterromanschreiber des deutschen Mittelalters den berühmten Satz schrieb, er könne keinen einzigen Buchstaben², zerging dieser scheinbare Widerspruch sehr einfach: Zum Lesen als Grundvoraussetzung seines Tuns war jener Dichter selbstredend imstande, zum Schreiben dagegen brauchte er einen Sekretär. Ganz entsprechend gibt es, zumal unter seinen heutigen Kollegen, nämlich den Computerkünstlern, manchen, der Computercodes zwar lesen und deshalb auch einsetzen kann, aber ihr Anschreiben sogenannten Programmierknechten überlassen muß.

Was schließlich mit denen geschieht, die Codes nicht einmal lesen können, ist in der Theorie klarer als in zukünftiger Empirie. Theoretisch verriet es schon die stehende Redewendung, derzufolge Benutzer, was immer sie an der Konsole tun, es »unter« einem bestimmten Betriebssystem wie Unix oder Windows tun: Der Computeranalphabet als solcher ist, mit anderen Worten, zum Subjekt oder Untertan einer Corporation geworden. Er unterliegt dem digitalen Code genauso massiv und undurchschaubar wie etwa seinem genetischen Code. Davor behütet auch der geläufige Alphabetismus der Buchstaben in keiner Weise. Wahrscheinlich haben die Mächte, die den militärisch-industriellen Fortschritt des Westens auslösten, die allgemeine Schulpflicht und damit das demokratische Lesen/Schreiben-Können ihrer Bevölkerungen ohnehin erst in dem historischen Augenblick eingeführt, als sie selber, mit der militärischen Geheimitelgraphie von 1794, über einen ersten Code verfügten, der digital war und nicht mehr alphabetisch. Schon deshalb macht es dem Reich der beherr-

2 Wolfram von Eschenbach, *Parzival* (hg. v. Karl Lachmann), S. 115, V. 27: »ine kan decheinen buochstap«.

schenden Zahlen keine Mühe, gerade in seiner Allgegenwart undurchschaubar zu bleiben. Informierte Zeitgenossen verkünden, neuerdings würden sogar analoge Chips entwickelt werden. Sie haben also vergessen oder nie gehört, daß integrierte Schaltkreise seit ihrer Entwicklung 1957 sämtlich analog gewesen waren, bevor ein großes Raketenprojekt des Pentagons vier Jahre später, bei Texas Instruments, den Startschuß zur Digitalisierung abgab. Die Technikgeschichte auf ihrem Triumphzug produziert mithin selber die Illusion, daß es andere Prinzipien als die in Digitalcomputern realisierten gar nicht geben könne. Was Wiener Shannons Wahnsinn nannte, ist als Verleugnung aller analogen Verfahren zum Industriestandard geworden. Aber just dieses verleugnete Analoge, so alt wie sonst nur Rauschen und Chaos, schwebt der Computerindustrie neuerdings als Allheilmittel zur Behebung ihrer sogenannten Softwarekrise vor. Den Computeranphabeten, die Codes weder lesen noch schreiben können, soll dadurch geholfen werden, daß sie mit binären Zahlen und unverständlichen Buchstabenfolgen überhaupt nicht mehr in Berührung kommen. Die Innereien der Maschine bleiben selbstredend weiter digital, weil sie sonst gar nicht laufen würde, aber ihre Benutzerschnittstelle nimmt mehr und mehr die Züge analoger Unterhaltungsmedien an, wie sie seit gut hundert Jahren vertraut sind. Unter dem Schlachtruf Multimedia wird es als bald eine Neuauflage von Grammophon – Film – Schreibmaschine geben, bei der die Schreib-Rechen-Maschine namens Computer ihre Benutzer nurmehr als analphabetische Augen und Ohren adressiert.

Die Stadien dieser Entwicklung zum Multimediensystem sind rasch aufgezählt. In den allerersten Geheimdienstjahren warf die Maschine ihre Ergebnisse als binäre Rohdaten aus. Wie im Fall von Shannons genialem Spielzeug fielen Design und Funktion zusammen. Konrad Zuse und Alan Turing dürften mithin die einzigen Menschen gewesen sein, die das Zweierzahlensystem genauso leicht wie unser Zehnersystem lesen können mußten und, im Fall Turing, darauf auch noch stolz waren (vgl. Hodges 1983: 399). Die nächste Computergeneration sah dagegen Möglichkeiten und (bemerkenswerterweise) erstmals Frauen vor, die der weiterhin blindtaubstummen Maschine Lochkarten von einiger Lesbarkeit eingeben konnten. Die Enkelgeneration fiel zusammen mit einem Fernsehzeitalter und folglich mit einem Bildschirm, dessen sogenannte Kommandozeile UNIX-Entwickler wie Kernighan und Ritchie sodann mit ihrem berühmten UNIX-Kauderwelsch programmieren durften: Heute noch sagt die Zeichenfolge: »dd if=/dev/rmt0 of=/usr/harry/prog\n« dem Betriebssy-

stem soviel wie: ›Kopiere den Inhalt des Magnetbandes rmt0 ins Verzeichnis der Programme des Benutzers Harry, wobei, ohne Abbruch bei möglichen Lesefehlern, aus IBM-Zeichen ASCII-Standardzeichen und aus Großbuchstaben Kleinbuchstaben werden sollen‹ (vgl. Drees 1988: 330). Derartig zeichenökonomische Zaubersprüche sind den Analphabeten, die seit einem Jahrzehnt mit Personal Computern beglückt werden sollen, selbstredend nicht mehr zuzumuten. Als Benoît Mandelbrot einen altersschwachen Tectronix-Bildschirm 1980 dazu mißbrauchte, statt eindimensionaler Kommandozeilen zweidimensionale Bilder oder Fraktale anzuzeigen, begann auch das Zeitalter graphischer Benutzeroberflächen. Das ehrenwerte Motiv hinter diesem Design, das mittlerweile schon zur Dreidimensionalität von Virtual Realities fortgeschritten ist, liegt auf der Hand: Je mehr Dimensionen eine Benutzerschnittstelle selber aufweist, desto mehr Systemzustände macht sie zugänglich und mithin auch steuerbar. Wenn heute ein einziger Chip, etwa der Intel 80586, 3 Millionen Transistoren oder 500 000 Schaltzustände integrieren kann, wird solche Transparenz zur bitteren Notwendigkeit.

Die Kurzgeschichte der Computerschnittstellen kehrt also im Eiltempo eines halben Jahrhunderts den langen Marsch um, als den Vilém Flusser die Zeichenpraxis der Gattung Mensch so gern beschrieben hat: Am Anfang ein vierdimensionales Kontinuum aus Raum und Zeit, nur mit dem Nachteil behaftet, daß keine Einzelheiten verarbeitet, übertragen, gespeichert werden konnten. Daraufhin, als Einführung von Codes im allgemeinen, die Herauslösung dreidimensionaler Klötze aus diesem Kontinuum, einfach um es bezeichnen zu können: mit einem Grabstein, einer Pyramide, einem Götterstandbild. Also ein erstes Symbolsystem mit dem einzigen Nachteil, daß jeder solche Klotz, einfach weil er da ist, etwas anderes notwendig verdeckt. Um dieses Handicap zu beheben, fährt Flussers Rekonstruktion fort, wurden die Klötze zunächst durch zweidimensionale Bilder abgelöst und diese Bilder, wann immer Bilderstürme oder Reformationen die ihnen eigene Verdeckung erkannten, ihrerseits durch lineare Schriften ersetzt. Schließlich und endlich wich die Verdeckung, die auch und gerade unsere Buchkultur aus schreibenden Göttern, Dichtern und Denkern produziert, einem Zeichensystem von null Dimensionen, das Verdeckungen folglich definitionsgemäß ausschließt: dem Zifferncode der Mathematik.

Mag sein, daß Flussers Rekonstruktion der Gattungsgeschichte allzu elegant läuft; als Umkehrmodell der Schnittstellenentwicklung von Com-

putern eignet es sich allemal. Denn beim Fortschritt von der nullten zur mittlerweile dritten Dimension, vom Lochkartenstapel zur graphischen Standardoberfläche, erhebt das Gespenst der Verdeckung aufs neue sein Haupt. Die geballte Rechenleistung heutiger Mikroprozessoren, die den Aufbau bewegter Graphiken überhaupt erst möglich gemacht hat, verschafft der Industrie offenbar vorrangig eine Gelegenheit, ihr alphanumerisches Wissen vor lediglich alphabetischen Kunden erfolgreich zu verstecken. Insofern dürfte den Computeranalphabeten in der Empirie auch weiterhin mehr Glück beschieden sein, als theoretisch zu erwarten.

Der Chip mit den 3 Millionen Transistoren zum Beispiel enthält einige hunderttausend, die im wörtlich so genannten Protected Mode keinerlei Rechenleistung erbringen, sondern nur diese Abschottung (vgl. Kittler 1993). So wird die Trennlinie zwischen einer neuen Elite und dem Rest der Welt zum integralen Teil von Hardware und Software gleichermaßen, also zementiert. Graphische Benutzeroberflächen entfalten genau dieselbe Menschenfreundlichkeit, die einst dem finsternen Mittelalter seine Armenbibeln aus lauter Bildern eingab. Damit die Fähigkeiten, Code zu schreiben und zu lesen, Monopol der Hersteller bleiben, befördert der Benutzer mit Hilfe seiner Mickey Mouse³ zehn Ikonen in den gleichfalls ikonischen Papierkorb, anstatt mit einer einzigen, aber alphabetischen UNIX-Kommandozeile die zehn entsprechenden Dateien zu löschen. Daß das Zeitverschwendung ist, räumt selbst das autoritative Handbuch über Graphikprogrammierung ein (vgl. Foley u.a. 1990: 398). Mehr noch, dieses Handbuch gelangt zu der ebenso bemerkenswerten wie paradoxen Einsicht, daß »einige Anwendungen, wie etwa das Programmieren, sich zur direkten [nämlich graphischen] Manipulation nicht von selber schicken«.⁴ Mit anderen Worten: Wo immer eine Anwendung die Kluft gegenüber der Systemprogrammierung einebnet, weil sie wie die sogenannte Anwendung namens Programmierung schlicht und einfach das tut, wofür Computer schließlich konstruiert worden sind, fällt mit den graphischen Benutzeroberflächen auch die künstliche Mauer zwischen Herstellern und Benutzern wieder dahin. Eine Maschine, die nach Turings Definition alle anderen Maschinen soll imitieren können, kann eben keine unveränderbaren Zustände einnehmen. Sie stellt den Begriff des geistigen Eigentums, an dem die Computerindustrie inständiger festhält als alle Schriftsteller,

³ Daß die Maus von Mickey Mouse abstammt, beweist ihre Maßeinheit Mickey.

⁴ Foley u.a. (1990: 398, Übersetzung F. K.).

die diesen Begriff seit Fichte und Goethe erfunden haben, grundsätzlich in Frage.

Sicher, ein technisch gezüchteter und zementierter Computeranalphabetismus wirft Geld ab: Leute, die Codes weder lesen noch schreiben können, werden unmöglich zu Hackern. Aber ob dieser elektronische Protektionismus, der allmählich aus den USA nach Brüssel sickert, sein Geld wert ist, steht in den Sternen. Als Präsident Bush alle fünfzig Gouverneure aller fünfzig Bundesstaaten in die akademische und das heißt schriftkulturelle Idylle von Charlottesville, Virginia, einlud, um über den Notstand des US-Erziehungssystems zu diskutieren, beklagten die Politiker einen neuen Analphabetismus, der auch High-School-Absolventen daran hindert, unter den eigenen Scheck noch die eigene Unterschrift zu setzen. Die Industrie dagegen beklagte etwas ganz anderes: einen Analphabetismus nicht im Reich der sechsundzwanzig Buchstaben, sondern auf dem weiten Feld zwischen Binärcodes und Programmiersprachen, Flußdiagrammen und Algebra.

Die Leute – also das, was einmal »Der Mensch« hieß – sind ohne ihren Maschinenpark eben nicht zu denken. Der klassische Facharbeiter starb in weiten Bereichen aus, als Henry Ford am Ersten Tag des Ersten Weltkriegs gleichzeitig die Fließbandarbeit und das Modell T einführte. Und das nicht etwa, um die Autoherstellung zu beschleunigen, sondern ganz ausdrücklich zu dem Zweck, Arbeiter einstellen zu können, die mit achtundvierzig Stunden Schulung auskamen. Heute stirbt umgekehrt der moderne Fordarbeiter in weiten Bereichen aus, weil die Computerisierung von Wissenschaft und Büroarbeit den übrigen Maschinenpark ja nicht verschont. Computer Aided Manufacturing, dieses unter Federführung von General Motors entwickelte industrielle Herstellungsverfahren, kommt ohne Computeralphabeten nicht aus. Auch Roboter müssen schließlich programmiert werden, in den Werkshallen selber und nicht nur auf Chefetagen. Deshalb das neue Anforderungsprofil der amerikanischen Industrie, deshalb ihr Ruf nach Flußdiagrammlesern oder BASIC-Programmierern – und das alles im Angesicht der Tatsache, daß die Schere zwischen Anforderungsprofil und Ausbildungsprofil Jahr um Jahr noch zunimmt. In den USA klappt diese Schere bekanntlich mit Vorliebe in den Niederungen, irgendwo zwischen High Schools und Slums. Hierzulande herrscht sie eher auf den Höhen der Schriftkultur. Wer wie Heiner Müller Turings Satz zitiert, daß die Computer mit Sicherheit übernehmen werden, dann aber fortfährt, wir sollten es ihnen nicht zu leicht machen (vgl.

Müller/Weimann 1991: 188), steht unter deutschen Schriftstellern recht einsam da. Europas grundlegende Unterworfenheit unter Buchstabenalphabeten bewegt Dichter und Denker noch immer dazu, andere Codes, statt ihnen auch nur Widerstand zu leisten, lieber gar nicht erst zu ignorieren. Bestenfalls trennt man zwischen Sprache auf der einen Seite und Technik auf der anderen, zwischen kommunikativem und instrumentellem Handeln, als führten nicht gerade die europäischen Schriften den Beweis, daß es keine Kommunikation ohne Nachrichtentechnik gibt. Ein Computeralphabetismus, der nur Heimweh ins Niedagewesene ist, steht also in Blüte. In seiner schönen Naivität ergänzt und vervollständigt er die Verkaufstaktik einer Software-Schmiede, die im Blick auf weitere Dollar-Milliarden ihr minderbemitteltes Betriebssystem neuerdings durch Graphik-Oberflächen für Analphabeten verschönert. Nur folgerichtig lautet denn auch der häufigste Satz, den Intellektuelle hierzulande über ihren häuslichen Computer kundtun: »Ich benutze ihn aber nur als bessere Schreibmaschine.« Genauso handelten einst, unmittelbar nachdem Gutenberg den Buchdruck mit beweglichen Lettern erfunden hatte, die Mönche eines Freisinger Klosters. Mönche hatten ihre Manuskripte ja Jahrhundert um Jahrhundert, Stück für Stück immer wieder abschreiben müssen. Also beschloß jenes Kloster, kaum daß ihm der Erstdruck seines Meßbuches gelungen war, alle vierhundert Exemplare Stück um Stück auf Abschreibefehler hin zu kontrollieren (vgl. Giesecke 1991: 415). Buch auf, Buch zu, Buch auf, Buch zu: ganz wie bei Shannons Spielzeug.

Nun besitzt aber Shannons Spielzeug, im Unterschied zu allen Maschinen der Vergangenheit, die seltsame Eigenschaft, Kinder in die Welt setzen zu können, die klüger als ihr Erzeuger sind. Fünfzig Jahre Elektronik haben diesen theoretisch längst bewiesenen Satz auch praktisch vorgeführt: Nur Computer sind überhaupt noch imstande, Computer schneller und besser als sie selber zu konstruieren. Deshalb schöpfen weder Schreibmaschinen für Schriftsteller noch multimediale Spielzeuge für Analphabeten die technischen Möglichkeiten von heute aus. Der lange Weg von Binärzahlen über Lochkarten und Kommandozeilen zu graphischen Benutzerschnittstellen zeigt vielmehr einen grundsätzlichen Trend an. Zur Beschreibung von Maschinen mit zwei Schaltzuständen wie Shannons Spielzeug war auch sein algebraischer Wahnsinn die einzig mögliche Methode. Maschinen aus Millionen von Schaltzuständen machen dagegen den desperaten Versuch, mit algebraischen oder diskreten Mitteln ihr genaues Gegenteil zu simulieren: analoge Prozesse, Differentialgleichungen, reelle

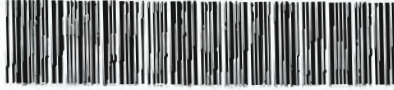
Zahlen. Ihre Schnittstellen stehen also nicht einfach oder ausschließlich den Benutzern offen, sondern gleichermaßen einem Zufallsreich, das einst Natur geheißen haben mag. Alle Sensoren und Effektoren, wie sie auf dem zögerlichen Weg von der Militärtechnik über die Industrie neuerdings auch im Personal Computer Einzug halten, zeugen davon. Mit dem Einbau analoger Schnittstellen aber fällt die Beschränkung weg, daß Computer in ihrer Umwelt immer nur Sekretärinnen, Aktenberge und das heißt jene Bürokratie antreffen, die die International Business Machines Corporation dem laufenden Jahrhundert beschert hat. Ab sofort begegnen sie vielmehr einem Analphabeten, der alle anderen in den Schatten stellt. »Dum deus calculat, fit mundus«, verkündete Leibniz, der vor drei Jahrhunderten auch den ersten Binärcode anscrieb: »Indem Gott rechnet, entsteht Welt.« Diesen Glauben an mathematische Gesetze und damit an einen alphanumerischen Code, dem die Natur im ganzen untertan wäre, hat auch der Atheist Alan Turing nicht erschüttert. Im Gegenteil, seine Theorie einer Universalen Diskreten Maschine, die alle anderen Maschinen imitieren kann, besagt in ihrer stärksten, nämlich physikalischen Form, daß die Natur selber eine Turing-Maschine sein muß. Erst in jüngster Zeit ziehen Chaostheorie und informatische Komplexitätstheorie genau diese Hypothese in Zweifel (vgl. Herken 1988). Wenn Wolken regnen oder Wellen entstehen, haust in ihren Molekülen wohl kaum ein digitaler Computer, der, indem er rechnet, das Wetter von heute und morgen machen würde. Die Gleichungsgesetze, mit denen Wetter für Ausschnitte der Zukunft allmählich vorhersagbar geworden ist, beschreiben vielmehr einen Analphabeten. Eben deshalb gibt es, was einst das Schöne und das Erhabene hieß. Wenn aber Turings Diskrete Maschine nicht universal ist, taucht die Frage nach den Grenzen ihrer Leistung auf. Schon sind die ersten mathematischen Schritte des Nachweises getan, daß die rasante Vermehrung von Transistoren pro Chipfläche, wie sie seit zwei Jahrzehnten als Allheilmittel läuft, grundsätzlich außerstande bleibt, mit der Komplexität rückgekoppelter Naturphänomene mitzuhalten. Der digitale Ansatz selber zieht Grenzen, die für analoge Computer einer allerdings noch vollkommen hypothetischen Architektur nicht gleichermaßen gelten würden. Womöglich ist der Computeralphabetismus also nicht das Ende von Geschichte überhaupt, wie Medienpropheten es so gern verkünden, sondern nur eine Etappe im Berechnen dessen, was turing-unberechenbar scheint.

Literatur

- Borland International (1989), *Turbo Debugger*, München.
- Drees, Horst (1988), *Unix. Ein umfassendes Kompendium für Anwender und Systemspezialisten*, Haar bei München.
- Foley, James D. u.a. (1990), *Computer Graphics, Principles and Practice*, Reading/Mass.
- Giesecke, Michael (1991), *Der Buchdruck in der frühen Neuzeit. Eine historische Fallstudie über die Durchsetzung neuer Informations- und Kommunikationstechnologien*, Frankfurt/M.
- Hagemeyer, Friedrich Wilhelm (1979), *Die Entstehung von Informationskonzepten in der Nachrichtentechnik. Eine Fallstudie zur Theoriebildung in der Technik in Industrie- und Kriegsforschung*, Diss., Berlin.
- Herken, Rolf (Hg.) (1988), *The Universal Turing Machine. A Half-Century Survey*, Hamburg/Berlin.
- Hodges, Andrew (1983), *Alan Turing: The Enigma*, New York.
- Hornung, Christoph/Pöpsel, Josef (1989), *3-D à la carte*, c't 10/1989.
- Kahn, David (1967), *The Codebreakers: The History of Secret Writing*, London.
- Kittler, Friedrich (1993), »Protected Mode«, in: *Draculas Vermächtnis. Technische Schriften*, Leipzig, S. 208 – 242.
- Lacan, Jacques (1966), »Ouverture de ce recueil«, in: *Ecrits*, Paris.
- Müller, Heiner/Weimann, Robert (1991), »Gleichzeitigkeit und Repräsentation. Ein Gespräch«, in: Weimann, Robert/Gumbrecht, Hans-Ulrich (Hg.), *Postmoderne – globale Differenz*, Frankfurt/M.
- Nietzsche, Friedrich (1967 ff.), »Jenseits von Gut und Böse. Vorspiel einer Philosophie der Zukunft«, in: *Werke. Kritische Gesamtausgabe*, hg. von Giorgio Colli und Mazzino Montinari, Bd. VI, 2.
- Pynchon, Thomas (1981), *V*, New York.
- Rolle, Karl-Hermann (1989), *Das Turbo Pascal 5.0 Buch*, Düsseldorf/San Francisco/Paris/London/Arnheim.
- Shannon, Claude E. (1938), »A Symbolic Analysis of Relay and Switching Circuits«, in: *Transactions of the American Institute of Electrical Engineers* 57, S. 713 – 723.
- Spengler, Oswald (1923), *Der Untergang des Abendlandes. Umriss einer Morphologie der Weltgeschichte*, Bd. 1, München.
- Wiener, Norbert (1962), *Mathematik, mein Leben*, Düsseldorf/Wien.
- Wolfram von Eschenbach (1926), *Parzival*, hg. von Karl Lachmann, Berlin/Leipzig.

Redaktion: Ralf Schilberg

UB LÜNEBURG LÜN4
0 382 415 2



BID 1625 002

(BID 040)

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Literatur im Informationszeitalter / Dirk Matejovski;
Friedrich Kittler (Hg.). – Frankfurt/Main; New York:
Campus Verlag, 1996

(Schriftenreihe des Wissenschaftszentrums Nordrhein-Westfalen; Bd. 2)
ISBN 3-593-35058-0

NE: Matejovski, Dirk [Hrsg.]; Wissenschaftszentrum Nordrhein-Westfalen
<Düsseldorf>; Schriftenreihe des Wissenschaftszentrums ...

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung ist ohne Zustimmung des Verlags unzulässig. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Copyright © 1996 Campus Verlag GmbH, Frankfurt/Main

Umschlaggestaltung: Atelier Warminski, Büdingen

Satz: Fotosatzstudio »Die Letter«, Hausen/Wied

Druck und Bindung: Friedrich Pustet, Regensburg

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Printed in Germany

UB Lüneburg

96 – 15404

Universität Lüneburg

— Bibliothek —

Inhalt

Vorwort der Herausgeber	7
Die Auswanderung der Zahlen aus dem alphanumerischen Code <i>Vilém Flusser</i>	9
Sprachnormen, Verrechtlichung und die Mediatisierung von Kommunikation <i>Rüdiger Weingarten</i>	15
– Zwei Kulturen im Zusammenstoß – Erkundungen im Medienland über ein Leseland <i>Hilmar Hoffmann</i>	– 37
Audiovisualität im Mittelalter <i>Horst Wenzel</i>	50
Späne, Kreuze, Initialen – Schriftzeichen als Beglaubigungsmittel in mittelalterlichen Tristan-Dichtungen <i>Urban Küsters</i>	71
Die Information in der Bibliothek der Gegenwart <i>Günter Gattermann</i>	102
Zur Virtualisierung von Bibliotheken und Büchern <i>Rainer Kuhlen</i>	112