

Frieder Nake

DER SEMIOTISCHE CHARAKTER DER INFORMATISCHEN GEGENSTÄNDE

Diskussion mit Hanna! – über Technik (laut Hanna) als Kniff, die Welt so einzurichten, daß wir sie nicht erleben müssen. ... Technik als Kniff, die Welt als Widerstand aus der Welt zu schaffen ...
Max Frisch, *Homo faber*

Was Max Frisch, der Mann und Dichter, seine Romanfigur Faber, den Ingenieur, ins Tagebuch schreiben läßt über eine Bemerkung von Hanna, der Frau, zur Technik, das gehört zum Erstaunlichsten und Einleuchtendsten. Technik als Kniff, als Kniff von Männern, als Kniff, mit dem diese begeisterungsfähigen und erfindungsreichen Männer sich Mittel schaffen, die Welt zu bewältigen, indem sie sie gewissermaßen beseitigen, zur Seite schieben, als unreine und undurchsichtige Vorlage dafür nehmen, eine künstliche, saubere und vorherbestimmte Welt zu schaffen. Hanna weist, wie könnte es anders sein, einen Augenblick später, wenn wir Fabers Tagebuchnotiz Glauben schenken dürfen, auf die unerbittliche Zeit hin, in der Leben zur *Gestalt* wird, also seine Endlichkeit zeigt, die jenseits der unablässigen Addition und Ausdehnung liegt, in der Technik sich ergeht.

Mit großem Vergnügen und melancholischem Verlangen greife ich oft zu dieser Passage in Frischs Text. So auch jetzt, wo ich mich anschicke, eine Bemerkung zu Papier zu bringen, die Elisabeth Walther aus Anlaß ihres Dreivierteljahrhunderts gewidmet sein soll. Man erlebt die Philosophin immer als nüchtern. Ganz bestimmt ist sie das in ihren Texten zur theoretischen Semiotik, in jenen formalistischen Äußerungen, in denen die Stuttgarter Schule sich niederschlägt, jene Schule, in deren Zentrum Elisabeth Walther stets neben Max Bense stand und der sie mit ihrer Allgemeinen Zeichenlehre das Lehrbuch geschrieben hat (Walther 74). Nüchtern ist wohl auch gewöhnlich ihr Vortrag, wenn auch sie selbst sich nicht als gar so nüchtern empfindet wie der Außenstehende es tun mag. Was mag sie sich denken, wenn ich in Gedanken an sie an Hanna erinnert bin?

Die Technik, zu der Hanna sich Faber gegenüber äußert, ist die Technik der Brücken, der Bauwerke, der Flugmaschinen, der Produktionsanlagen – das, woran wir alle gewöhnlich und jedenfalls bis vor kurzem denken, wenn von "Technik" die Rede ist. Diese Technik schlägt sich nieder, ja: existiert, in handfesten, bewegungsstarken, schweren, großen, empfindlichen und empfindbaren Manifestationen des menschlichen Geistes, des Faustischen Willens der Veränderung der Welt.

Welt ist heute längst und total technische Welt (wer hätte das klarer betont als Max Bense?). Nichts, was wir erleben von der Welt, die wir uns romantisch als technikfrei träumen, ist heute erlebbar *ohne* Technik. Und sei es auf so perverse Art, daß wir uns in eine schöne, einsame

Gegend flüchten, indem wir eine Flugmaschine besteigen. Wir kämen ohne Technik nie dorthin und erlebten somit auch nicht die Natur so pur, wie wir sie uns träumen, in ihrer Widerständigkeit, von der Hanna schwärmt.

Und doch hat Hanna mit ihrer Bemerkung, die sie Faber gegenüber anscheinend nicht weiter erläutern kann – muß sie nicht zugeben, daß sie nicht recht weiß, was sie damit meint? –, und doch hat Hannas Bemerkung getroffen: “Die Welt so einzurichten, daß wir sie nicht erleben müssen”. Zwar scheint Faber sich nach wie vor überlegen zu fühlen, Hanna kann ja ihre Empfindung nicht in Worten präzisieren. Doch er wird immerhin ein wenig nachdenklich.

Selbst wenn wir Hannas etwas traurige Kommentierung außer Acht lassen, Fazit ist: Welt läßt sich, unter dem Zugriff des Ingenieurs, *einrichten!* Und sie *wird* eingerichtet. Wieviel mehr muß uns das heute als zutreffend erscheinen, da der Technik eine weitere Schicht gewachsen ist, mit der es gelingt, nun auch noch die Technik selbst als verbliebenen resp. neu erst entstandenen Widerstand aus der Welt zu vertreiben. Ich spreche von der Informationstechnik und von der Informatik als ihrer wissenschaftlichen Disziplin. Wir wollen sehen, was der Charakter der Gegenstände dieser postmodernen Technik ist.

Die Moderne als Projekt der Semiotisierung

Wir können die Moderne auffassen als die allmähliche Verwandlung von Welt in Zeichen, erscheinen uns doch ihre vielfältigen Versuche, Welt theoretisierend zu erfassen und zu begreifen, als ein grandioses Projekt der Semiotisierung all dessen, was ist. Wo immer der moderne Mensch seinen Blick hinrichtete, schuf er eine Zeichenwelt jenes besonderen Aspektes von Welt, auf den er es abgesehen hatte. Jede spezifische Wissenschaft, von der Theologie bis zur theoretischen Physik, können wir begreifen als die Etablierung einer besonderen Zeichenwelt, mit deren Zeichen die erste Welt unter dem gewählten Aspekt zum Modell wird.

Daß der Mensch als tätiger Mensch die Zeichenwelt dann wieder in Maschinerie und künstlicher Welt stofflich verdichtete, ist ebenfalls wahr (Hannas Thema!), ändert jedoch nichts am Projekt. Naturgemäß ist dessen Vorgeschichte in den Religionen zu sehen, in denen die unerfahrbare Ganzheit der Welt, nach deren Erfahrung der Mensch sich so sehr streckt, zum beschwörenden Zeichen wird. Auch diese Tatsache ändert nichts am Projekt der Moderne, *bewußt* Welt der Semiotisierung zu unterwerfen, um sie gefügig zu machen.

Nehmen wir die Mathematik als Beispiel! Bei ihr haben wir es, betrachten wir sie nur ganz bei sich selbst, mit den reinsten Zeichen zu tun, mit Zeichen nämlich, die ihre Objekte immer schon selbst setzen.

„Sei $\epsilon > 0$ und $\chi \in A \dots$ “, sagt der Mathematiker und sieht in solchen Namen eine Welt vor sich, wo andere nichts erblicken als Kreidehäufchen auf der Tafel. Beobachten wir zwei Mathematiker ins fachliche Gespräch vertieft, so greifen sie nach kurzem zu Papier und Bleistift,

um ihre Zeichen vor sich zu haben. Auf Papier *schaffen* sie ihre Gegenstände. Sie sehen sie an und stellen sie hin, indem sie sie herstellen.

In der Informatik nun wird die Welt der Mathematik vorausgesetzt. Sie wird *gedacht*, ganz so wie Hilbert seine Geometrie beginnen läßt: "Wir denken drei verschiedene Systeme von Dingen ..." (Hilbert 62). Nicht: wir denken *an* drei Systeme, sagt er. Sondern: wir denken sie. Das heißt, wir schaffen sie im Denken. Sie sind nicht vorher und unabhängig vom Denken da, die Punkte, Geraden und Ebenen. Sie sind nur im Denken da, und Denken wird zu einer Form des Schaffens. Es schlägt sich nieder, so Hilbert, in Namen, die den gedachten Dingen gegeben werden, oder: als die sie existieren. Unterschiedliche Namensschemata, unterschiedliche Repräsentamina also, geben unterschiedliche Dinge (Punkte, Geraden, Ebenen). Wir denken Systeme von Dingen und schaffen sie in Zeichen. Repräsentamen und Objekt fallen in eins.

Hilberts Standpunkt mag vielen als absurd erscheinen. Denn wenn auch niemand jemals einem Punkt begegnet in der stofflichen Welt, so gibt es doch reale Anlässe genug, in abstrahierender Weise den Punkt zu entdecken bzw. eben zu setzen. Aber als Standpunkt ist Hilberts Denkfigur durchaus brauchbar. In diese Welt der gedachten Dinge hinein bringt die Informatik Bewegung. Wenn die Mathematik ihre Dinge als Zeichen höchster Abstraktheit schafft, so verleiht die Informatik ihnen eine Beweglichkeit, als seien sie stofflich. Wenn die Ingenieurdisziplinen klassischer Prägung in der Maschine die Welt hinter sich lassen und ersetzen, so macht die Informatik die Zeichen selbst zu Maschinen, die in relativer Autonomie auf die klassischen Maschinenprozesse einwirken. Nicht Zeichen *für* Maschinen, die es zu bauen gilt, denen das Zeichen *Entwurf* ist, sondern Zeichen *als* Maschinen – so treibt die Informatik in ihrem Konzept der Berechenbarkeit die moderne Semiotisierung auf die Spitze. Träfe Hanna einen von diesen Ingenieuren, was würde sie sagen? Das gilt es näher zu erläutern, indem ich versuchen werde, den semiotischen Charakter der Gegenstände der Informatik aufzuzeigen.

Software

Womit hat es die Informatik zu tun? Was auch immer der Anlaß und das Ziel der Arbeit eines Informatikers oder einer Informatikerin sei, sie hat es mit Software zu tun. Informatikerinnen! schaffen Software. Software besteht aus Programmen und Daten. In Programmen und Daten schaffen sie Informationssysteme. Das sind komplexe Systeme, die für komplexe Systeme stehen, oder die Teil von solchen sind.

Man wird einwenden: Hardware! Richtig – die ganze schöne Software taugt nichts ohne den Computer aus Schaltungen und Chips und Stromversorgung und immer auch noch etwas Mechanik und Fernsehapparat dazu. Der Computer muß, und koste es das Leben, der Software untergeschoben werden, soll sich ihre informatische Natur enthüllen.

Und doch falsch! Zu tun hat es die Informatik in ihrem Kern mit Software, indem sie den Computer voraussetzt, so wie sie auf der anderen Seite die algebraischen Strukturen der Mathematik voraussetzt. Den Computer stellt die Physik hin und die Elektrotechnik her. Und

naturgemäß besteht eine Wechselwirkung zwischen Hardware und Software. Doch die ausufernden Softwaresysteme aller Schichten, die sich als abstrakte Maschinen, wie man sagt, über der Hardware erheben, bilden den Kern informatischen Tuns. In den Softwaresystemen feiert es seine Triumphe, in ihnen nistet die künftige Hardware.

Wir können Software sehen und wir können sie gleichzeitig nicht sehen. Wir wissen, damit sie so wirkt, wie sie wirken soll, muß sie dem Speicher eines Computers einbeschrieben sein. In diesem Zustand sehen wir sie nicht. Wir sehen vielleicht den Speicher, von dem wir wissen oder annehmen, die Software befindet sich in ihm. Sie selbst jedoch bleibt unsichtbar und zeigt sich nicht selbst, sondern nur ihre Wirkung.

Wir können sie aber dennoch auch sehen, jedenfalls sichtbar machen, wenn wir sie einer anderen Software anvertrauen, die für die Wiedergabe auf einem Drucker oder Bildschirm sorgt. Die Software selbst aber ist es offensichtlich nicht, die sich da zeigt. Sie bleibt im Speicher verborgen und sendet uns nur eine Repräsentation. Der können wir, so wissen wir, sogar unterschiedliche Form geben, ohne ihren Inhalt zu ändern. Software ist also merkwürdig immateriell: materiell im Speicher vorhanden, doch finden wir im gleichen Speicher gleich darauf andere Software; und materiell auf dem Druckerpapier gezeigt, doch in einer letztlich zufälligen Form nur, die auch anders sein kann. Wenn wir dann sagen: "das ist meine Software" und dabei auf die sichtbare Wiedergabe deuten, so ist das, worauf wir zeigen, Druckerschwärze und nicht Software. Doch solcher Lapsus ist zu verschmerzen. Jedenfalls ist es klug, diese kleine Lüge zu ignorieren.

Software also besitzt, das zeigt bereits diese einfache Beobachtung, einen quasi immateriellen Charakter. Sie *ist* nicht immateriell, das wäre eine abenteuerliche Vorstellung. Doch sie kann, ohne ihre Bedeutung oder ihre Funktion zu ändern, in durchaus unterschiedlicher Form erscheinen.

Sie besitzt mit anderen Worten die erstaunliche Qualität, daß man sie benutzen kann, ohne sie auch nur im geringsten materiell zu verschleifen. Sie nutzt sich physisch nicht ab, nur moralisch, wie Marx sagen würde². Heute hat diese Eigenschaft längst zu einem widerwärtigen ökonomischen Kniff geführt. Software wird nicht erst in guter und bester Form auf den Markt gebracht, sondern bereits in unausgereifter, vorläufiger. Die Vorläufigkeit ist an der Nummer der Version ablesbar, die dem Namen der Software angehängt wird und die ein wichtiges Merkmal im Streit um Qualitäten geworden ist. Die aktuelle Version einer Software trägt immer schon den Zwang zur nächsten Version in sich.

Der Zeichencharakter von Software

Doch zurück! Im quasi-immateriellen Charakter von Software zeigt sich ihre semiotische Natur. Denn was da so erscheint, als sei es immateriell, und doch nicht leugnen kann, unabhängig von uns, also stofflich zu sein, das kann gar nichts anderes sein als *Relation*. Im relationalen

Charakter von Software liegt ihre Zeichenhaftigkeit. In Relation zueinander stehen in Software: der Code, eine Menge von Prozessen und Einbettungen dieser Prozesse in praktische Kontexte.

Erstens. Ohne eine bestimmte Beschreibung ("Code") existiert Software nicht. Der Programmtext und die Codierung der Daten sind hiermit gemeint. Dieses stoffliche Substrat stellt den dinglichen Aspekt von Software dar. Der Code ist stets eine endliche Beschreibung, die einem konventionellen Schema folgt (gewöhnlich eine Programmiersprache).³

Zweitens. Der Programmtext bezeichnet eine i. a. unendliche Menge von Prozessen. Jeder dieser Prozesse entsteht aus einem "Lauf" des Programmtextes. Ein Lauf ist die Abfolge von Zuständen, in die eine komplexe Maschine (ein Computer) gerät, wenn sie den Programmtext zusammen mit eventuell geforderten Eingabedaten abarbeitet. Die Struktur des Programms sorgt in Wechselwirkung mit den eingegebenen Daten dafür, daß die Menge dieser bezeichneten Prozesse unbeschränkt sein kann und i. d. R. auch ist. Das Geheimnis des Programmierens besteht geradezu darin, daß in Programmen endliche Notationen für unendliche Mengen komplexer Prozesse gefunden werden – wobei der Witz darin liegt, daß diese Prozesse maschinell ausgeführt werden können.⁴

Da diese Prozesse als dynamische Beiordnungen zum statischen Programmtext als Läufe eines Computers entstehen, sind sie berechenbar.⁵ Das heißt, daß sie mindestens prinzipiell durch berechenbare Funktionen beschrieben werden können. Jeder konkrete Prozeß entspricht einer bestimmten Auswertung der berechenbaren Funktion. Erlaubt oder verlangt der Programmtext interaktive Eingriffe einer Benutzerin, so wird an dieser Stelle die Berechenbarkeit suspendiert. Statt durch *eine* berechenbare Funktion werden die möglichen Prozesse dann durch eine komplexe Struktur von berechenbaren Funktionen beschrieben. Der Monolith des klassischen Programms löst sich auf in einen Steinbruch.

Drittens. Ohne daß ein Mensch Interesse an den Verläufen oder Ergebnissen der durch einen Programmtext beschriebenen Prozesse hätte, würde der Computer nicht dazu gebracht, einen solchen Prozeß auszuführen. Das Interesse des Benutzers erschöpft sich aber nicht in dem berechenbaren Prozeß als solchem – im Gegenteil, der Benutzer bettet den Prozeß und seine Ergebnisse immer sofort in seine besonderen praktischen Kontexte ein. Der automatisch ablaufende Prozeß ist nichts als eine Folge von maschinellen (nämlich elektromagnetischen) Zuständen. Diese sind dem Benutzer aber mehr als gleichgültig. Sein Interesse gilt den Gegenständen, Aussagen, Werten, Bildern, Vorstellungen, Schlußfolgerungen – was immer es sei, das in der geradezu zwanghaften, jedenfalls zwangsläufigen Einbettung der elektromagnetischen Zustandsfolgen in die Lebens- und Arbeitswelt des Benutzers sofort aufscheint.

Diese Art von Dreiheit also wird in Software gestiftet. Kurz können wir sagen: Software ist eine Dreiheit aus statischem Programmtext, unendlicher Menge dynamischer Prozesse und interessegebundener, praxisgeleiteter Einbettung in Zusammenhänge. Wir erkennen in dieser dreistelligen Relation das Repräsentanten, das Objekt und den Interpretanten des Peirceschen Zeichens.

Die besondere Relationalität von Software erweist sich als die *semiotische* Relation Peircescher Tradition: ein Erstes steht für ein Zweites vermittelt und zum Zwecke eines Dritten. Software als informatischer Gegenstand besitzt damit einen durchaus anderen Charakter als die Gegenstände klassischer Ingenieurarbeit.

Ihr anderer, transklassischer Charakter beruht darauf, daß in Software eben *nicht* ein *Ding*, sondern ein *Verhältnis* zu konstruieren ist. Vielleicht zu Recht ziehen es deswegen auch viele vor, im Falle von Software lieber von *Gestaltung* als von Konstruktion zu sprechen.⁶

Wenn Software zu entwickeln bedeutet, ein Verhältnis zu gestalten, so gelingt das nur in dem Maße, wie die drei Aspekte der genannten Relation tatsächlich ins Blickfeld gezogen und je einzeln auch gestaltet werden: der statische Programmtext rührt an die Frage *korrekter* Konstruktion; die Menge dynamischer Prozesse nimmt die Frage der *Effizienz* der berechenbaren Funktion auf; die Einbettung in praktische Zusammenhänge betrifft die *Benutzbarkeit* und *Nützlichkeit* der Software als System in Lebenszusammenhängen. So haben wir die drei großen Bereiche informatischen Tuns hier systematisch an ihrem Gegenstand festgemacht.

Zeichen auf der Benutzungsoberfläche

Wir können den Zugang zum semiotischen Grundcharakter der informatischen Gegenstände aber auch noch auf andere Weise finden.⁷ Ganz oberflächlich betrachtet, *zeigt* sich Software an der Peripherie des Computers. Dort wird durch visuelle, akustische und vielleicht schon bald auch noch andere sinnlich wahrnehmbare Signale Meldung von dem einen oder anderen der intern erreichten Zustände gemacht. Die Licht- oder Tonsignale auf der Benutzungsoberfläche haben offensichtlich Zeichencharakter.

Wir sehen da z. B. ein kleines Bildchen, dessen Aussehen an einen amerikanischen Aktenordner ("Folder") erinnert. Dieses insofern ikonische Zeichen steht nun aber keineswegs für einen irgendwo vorhandenen stofflichen Aktenordner. Vielmehr wird damit eine Datei bezeichnet, also ein Teil der Software, mit dessen Hilfe Daten zu einer strukturierten Einheit zusammengefaßt werden. Erst über eine vielleicht nützliche, didaktisch gemeinte Metapher wird unser Zeichen zum Ikon. In Wahrheit, als Zeichen für die Datei genommen, ist es ein Symbol. In seiner Verwendung innerhalb der interaktiven Software-Benutzung besitzt es zusätzlich eine indexikalische Komponente. Für uns als Benutzer aber ist das ziemlich belanglos, da wir die Situation sofort und ständig ganz pragmatisch betrachten, weswegen wir denken und sagen: "da finde ich ja meine Aufsätze zur Semiotik" – und meinen damit nichts anderes, als daß wir die Dateien identifiziert haben, in denen sich bestimmte Texte codiert wiederauffinden lassen. Wir nehmen das Zeichen in seiner physischen Erscheinung für den bezeichneten Gegenstand. Ungenau, wie unser Sprechen und wohl auch Denken hier ist, ist es doch nicht schädlich.

Das wäre leicht ausführlich in vielen Verästelungen darzulegen. Wir wissen: Software erscheint an der Benutzungsoberfläche als Zeichen. Benutzungsoberflächen sind neuartige, sehr komplexe, dynamische Zeichen. In ihrem Kern behalten diese komplexen Zeichen vieles von dem, was

eine Druckseite, ein Plakat, ein Bild auszeichnet. Doch dieses statische Moment ist nur eingebettet in eine Dynamik⁸, die teilweise die Aufmerksamkeit der Betrachterin steuert, teilweise nur Verzierung ist. Am deutlichsten ist das aktuell wohl im tastenden Aufkommen einer visuellen Gestaltung von *Web*-Seiten zu sehen. Das Medium drückt dieser Gestaltung z. B. die Aufgabe auf, an den immer erneuten Vorgang des Aufbaus des *Web*-Dokuments auf einem Bildschirm nicht unerhebliche Gestaltungsentelligenz zu verwenden. Es kann kaum verwundern, daß Gestalter alter Schule (und vielleicht auch neuer) sich gelegentlich über die Semiotik einen Zugang zum *Interface Design* schaffen (Mullet & Sano 95, Bonsiepe 96, Bürdek 91).

Mißverständnisse um die Maschinisierung

Tief im Kern der Informatik stoßen wir auf einen weiteren Zugang zum semiotischen Wesen der informatischen Dinge. Informatik hat es wesentlich zu tun mit der Maschinisierung von Kopfarbeit (Nake 92). Diese Aussage ist so selbstverständlich und naheliegend wie ungeliebt und mißverstanden, daß sich der aufgeklärte Mensch nur wundern kann.

Im Mißverständnis Nummer eins, das unsere Formel von der Maschinisierung von Kopfarbeit umgibt, wird sie als eine normative Setzung gelesen, als eine Zielformulierung. Beliebt ist dieses Mißverständnis unter linken, kritischen Köpfen, von denen es in der Informatik nicht mangelt. Nichts aber liegt ferner, als eine analytische Kennzeichnung gleich als Zielsetzung mißzuverstehen. Die kritische Betrachtung eines Gegenstandes führt bekanntlich immer zunächst zu einer schärferen, klareren, differenzierten Kennzeichnung des Gegenstandes im Begriff. So auch hier, zumindest der Absicht nach.

Mißverständnis Nummer zwei rührt vom Haß auf die Lohnarbeit und ihre Merkmale her. Es ist ja schön, daß viele Menschen aus ihrem Entsetzen gegenüber den Abträglichkeiten der Lohnarbeit kein Hehl machen. Daß ihnen aber jeder Funke von Utopie soweit verloschen zu sein scheint, daß sie "Arbeit" nur noch mit "Arbeit unter kapitalistischer Form", also mit Lohnarbeit gleichsetzen können, stimmt nachdenklich.

Mißverständnis Nummer drei wird durch das Wort "Kopfarbeit" verursacht. Man setzt sie gleich mit höheren, kreativen oder sonstwie positiv gedachten Formen von Arbeit. Vor allem aber denken solche Interpreten an individuelle Arbeit. All das geht völlig am Begriff vorbei. "Kopfarbeit" meint nämlich nichts anderes als die Planung, Leitung und Kontrolle anderer Arbeit. Kopfarbeit ist damit Arbeit an der Arbeit anderer. Sie besteht selbstredend aus geistigen und manuellen Anteilen, weil die ja auf der Ebene des einzelnen Arbeitenden niemals zu trennen sind. Kopfarbeit ist jedoch darüber hinaus von vornherein gesellschaftliche Arbeit. Sie tritt als eigenständige, abgetrennte Form von Arbeit erst auf relativ hoher Entwicklungsstufe der Arbeit auf, dann nämlich, als deren beide Momente: Arbeitsteilung und Kooperation, aus ihrer naturwüchsigen Entwicklung zu bewußten Instrumenten der Steigerung der Produktivkraft der Arbeit werden. Nachzulesen ist das selbstverständlich bei Karl Marx (Marx 70) und Alfred Sohn-Rethel (Sohn-Rethel 72).

Kopfarbeit nun, so verstanden, tritt als separate Arbeit in großem Maßstab auf, als die Manufaktur zur Fabrik wird und ganz besonders im Taylorismus (Taylor 95). Naturgemäß entgeht sie nicht der Anwendung ihrer eigenen Prinzipien auf sich selbst. Sie unterliegt selbst der Rationalisierung und dem gnadenlosen Verdikt des Kapitals zur Maschinisierung. Mit dem Computer war die dafür notwendige Maschinerie gefunden.

Maschinisierung ist nichts anderes als die Übertragung von Elementen der Arbeit auf Maschinen. Bei diesem Vorgang kann es nicht anders zugehen, als daß formalisiert und standardisiert wird. Das hat eine wunderliche Wirkung. Elemente von Expertenarbeit werden, wo ihre Formalisierung gelingt, über die Maschine tendenziell jedem verfügbar. Man braucht die Fähigkeit zu solcher Expertenarbeit nicht mehr selbst zu erwerben, denn sie ist per Knopfdruck abrufbar. Damit kann tendenziell jeder manches von dem erreichen, was bis dahin einen Experten verlangte. Wird man das vernünftigerweise als eine Form von Demokratisierung (per Verallgemeinerung) bezeichnen, so geht solche Demokratisierung jedoch einher mit einer Banalisierung. Denn die per Maschine abrufbare, per Knopfdruck in Aktion versetzbare Expertise ist nun naturgemäß keine solche mehr. Was einen Knopfdruck (oder auch das Nachschlagen in einer Gebrauchsanweisung) verlangt, verlangt eben keinen Experten.

All das ist nichts Besonderes, sondern alltäglicher Gang der Dinge und hier nur erwähnt, um der leidvollen Erfahrung mit dem Mißverständnis vorzubeugen, das bei der Nennung der "Maschinisierung von Kopfarbeit" bei ihrem Namen so bereitwillig von ansonsten klugen Köpfen produziert wird.

Wenn es die Informatik in ihrem Kern wesentlich mit der Maschinisierung von Kopfarbeit zu tun hat, so heißt das, daß Software entwickelt wird, die Kopfarbeit aufsaugt und ausspuckt. Vorgängig muß Kopfarbeit dafür beschrieben, formalisiert und algorithmisch gefaßt werden. Das sind die drei Transformationen, denen alle Gegenstände unterworfen werden, wenn sie der Informatik unterworfen werden. Ich nenne sie die semiotische, die syntaktische und die algorithmische Transformation.

Zeichen – unser Thema – begegnen uns hier nun in doppelter Form. Zum einen sind die Gegenstände, die Kopfarbeit unmittelbar hervorbringt, semiotischer Natur. Die Arbeit, auf die Kopfarbeit sich bezieht, muß ja repräsentiert werden, um Gegenstand von Arbeit werden zu können. Denn wenn Kopfarbeit auch Arbeit an der Arbeit anderer ist, so bearbeitet sie jene andere Arbeit doch aus der Ferne, muß sich also ein vorausschauendes oder ein beobachtendes Zeichen davon machen. Banal ausgedrückt, schlägt Kopfarbeit sich in Texten, Plänen, Schemata, Zeichnungen, Anweisungen etc. nieder – also in semiotisch geprägten Gegenständen.

Zum anderen aber muß Kopfarbeit, soll sie maschinisiert werden, selbst repräsentiert werden. Denn sie, deren Gegenstand die Arbeit anderer ist, wird in ihrer Maschinisierung zunächst zum Gegenstand wieder anderer, nur um schließlich als Software zu kristallisieren. So haben wir es bei der Maschinisierung von Kopfarbeit mit einer doppelten, verschränkten Semiose zu tun. Die Gegenstände der betreffenden Kopfarbeit sind als Zeichen zu geben und die Vorgehensweisen der Kopfarbeit ebenfalls. Nicht verwunderlich ist, daß die größten Erfolge dieses

historischen Prozesses in den allgemeinsten Bereichen von Kopfarbeit zu finden sind: das Sortieren und Suchen nach beliebigen formalen Ordnungsschemata, Textbearbeitung, Tabellenkalkulation.

Die semiotische Natur von Software zeigt sich bei der Betrachtung der Maschinisierung von Kopfarbeit als ein fortgesetzter Prozeß der Verdoppelung. Wann immer wir zu einem Ding oder Vorgang eine Beschreibung, also ein Zeichen schaffen, schaffen wir ein Doppel des Dings oder Vorgangs. Kein identisches Doppel, selbstredend, sondern eines, das unseren weiteren Zwecken entgegenkommt, d. h. formalisierbar ist und algorithmisch gefaßt werden kann.

Technische Semiotik

Im Zeichen liegt nun, anders als im Ding, stets und offensichtlich eine ungeheure und doch leicht zu habende Entwicklungskraft: seine prinzipielle Rückbezüglichkeit. Sobald wir die Welt der separaten und erratischen Dinge verlassen und die der Relationen betreten, öffnen wir in eine Unendlichkeit. Ein Ding ist ein Ding für sich und bei sich. Ein Zeichen aber steht für anderes und ist deshalb immer schon in Prozesse eingeflochten. Diese Prozesse machen naturgemäß vor dem Zeichen nicht Halt und der Interpretant, die wichtigste Seite des relationalen Zeichens, ist selbst wieder Zeichen. Der Begriff des Zeichens selbst ist rekursiv und das einzelne Zeichen, das immer schon gemeinsam mit anderen Zeichen nur vorkommt, wird rekursiv im Prozeß.

Die Rückbezüglichkeit ist aber auch die vornehmste technische Eigenheit von Software. Daten können Programme, Programme können Daten werden. Funktionen können rekursiv definiert werden und Datenstrukturen auch. Es ist nicht übertrieben, die Informatik als die Erforschung der maschinell dargestellten Rekursivität zu verstehen. Die aber ist von zutiefst semiotischer Natur. Wenn wir Informatik überhaupt an ihrem Gegenstande fassen wollen – und das müssen wir, wollen wir sie als Disziplin begreifen – so müssen wir das semiotische Wesen von Software ins Zentrum rücken. Informatik erweist sich dann als *Technische Semiotik*, d. h. als eine Wendung der Semiotik hin zur Konstruktion von Semiosen in maschinellen Abläufen, nicht nur zu deren Interpretation.

In einem Atemzug ist der Computer dann als *semiotische Maschine* zu kennzeichnen. Den Begriff hat erstmals Nadin 1977 verwendet. Er ist später z. B. von Coy wieder aufgegriffen worden. In unserer Interpretation besagt er mehr als das häufig anzutreffende Wort von der symbolverarbeitenden Maschine. Der Computer ist semiotische Maschine insofern, als er *die* Maschinenklasse konstituiert, die notwendig ist, um Zeichenprozesse in syntaktisch-algorithmisch verkürzter Form zwar, ansonsten aber überhaupt zum Gegenstand technischer Konstruktion zu machen. Es gelingt eben nur mit dem Computer (aufgefaßt als Turing-Maschine), daß einer konkreten Maschine die abstrakte Beschreibung einer beliebigen anderen Maschine gleicher Art eingegeben wird, woraufhin sie sich so verhält wie jene andere, beschriebene. Einer klassischen Maschine kann man solches oder ähnliches durch Eingabe irgendeines ihr zuträglichen Materials nicht antun.

Mancherlei folgt von hier mit Zwangsläufigkeit. So zunächst die Flüchtigkeit der informatischen Gegenstände, die zur besonderen Faszination von Software, Hardware und Informatik unter Jugendlichen und Lehrenden führt. Sie wiederum ermöglicht das – im Rahmen interaktiver Berechenbarkeit – freie Simulieren modellierter Welten, also die Vertauschung von aktueller mit virtueller Realität. Der Computer erweist sich als die Maschine des Als-ob, die sich derzeit als instrumentales Medium enthüllt (Schelhowe 97), dessen Kontur und Reichweite erst allmählich Form annehmen.

Das alles muß naturgemäß Konsequenzen für die Gestaltung der informatischen Gegenstände aufwerfen. So ist denn auch zu beobachten, daß Gestaltung von Software sich allmählich der Tatsache bewußt wird, daß im Prinzip ein jedes Zeichen auf jedem Computer anders erscheinen kann als auf dem vorherigen. Gestaltung von Dokumenten für das WWW sieht sich einer Flüssigkeit und Biegsamkeit ausgesetzt, die klassischem Produktdesign fremd ist.

Der semiotische Charakter der informatischen Gegenstände darf uns über ein ganz wesentliches Moment nicht hinwegtäuschen: Was dem Menschen Zeichen ist, ist dem Computer Signal. Damit meine ich folgende Tatsache. Sobald ein Zeichen in der Form von Software in einen Computer eingeht, wird es reduziert auf seine syntaktische Komponente. Es verliert damit seine eigentliche semiotische Natur, wird auf seinen stets vorhandenen dinghaften Anteil zurückgestutzt. Wenn das Zeichen in seinem Charakter als *Zeichen* ständig einer prinzipiell unbegrenzten Interpretierbarkeit offen ist, weil das nämlich das ganze Geheimnis seiner pragmatischen Natur ist, so ist es umgekehrt als *Signal* in seiner Reduktion auf Syntaktik aller Interpretierbarkeit beraubt. Es ist dann das, was es als Repräsentamen ist, sonst nichts. Es weist, als Signal, nicht mehr über sich hinaus, sondern nur auf sich selbst hin. Nehmen wir das Signal wahr, so können wir diese Reduktion zwar nicht wirklich nachvollziehen, weil wir als semiotische Tiere (Hausdorff) einem andauernden Zwang zum Schaffen von Interpretanten unterliegen.

Doch in der Maschine ist das anders. Dort wirkt das als Signal harmlose Zeichen segensreich, indem es genau jene Funktion in einem automatischen Ablauf anstößt, für die wir es ausersehen hatten. Wir *wollen* gar nichts anderes erreichen, und es bleibt das Geheimnis der Apologeten einer Künstlichen Intelligenz, aus welcher Tiefe des Bauches der Maschine sie deren behauptete Fähigkeit zur Schaffung von Interpretanten heraufziehen sehen – und eben darin leugnen, daß, wie auch immer die Signale auf der Oberfläche eines Computers flimmern und piepsen, *ich* es bin, der anderes als Flimmern und Piepsen darin sieht bzw. dort hineinlegt.⁹

Der Wechselprozeß zwischen einem Menschen und einem Computer *erscheint* uns manchmal wegen der semiotischen Natur der hierbei verwendeten Gegenstände als symmetrisch und partnerschaftlich. Er *ist* es mitnichten. Während *wir* im Prinzip mit jedem Zeichen jeden beliebigen Kontext verfügbar haben und machen können, zeichnet sich die Maschine in ihrer Reduktion der Zeichen auf Syntaktik durch eine maximale Dekontextualisierung aus. Von allen Kontexten nämlich wird im Computer abgesehen außer von einem einzigen und letzten: von der Quantität. In der bloßen Quantität sind uns die Dinge nur als unterscheidbare vorhanden. Die Unterscheidung und nichts sonst ist aber auch der Inhalt der Zeichen, die der Computer

kalkuliert. Das Frappierende der Angelegenheit liegt darin, daß diese ärmlichen und deswegen kalkulierbaren Zeichen, die Signale, doch gleichzeitig zu kalkulierenden Zeichen werden.¹⁰

In reiner Quantität zu denken, maximal dekontextualisiert, das ist *Rechnen*. Rechnen ist die letzte, allgemeinste Form des Denkens. Sie ist so stark reduziert, daß sogar Maschinen konstruiert werden konnten, die sie meistern.

Das Wunderbare ist nicht, daß es Maschinen gibt, die an einer Form des Denkens¹¹ partizipieren. Das Wunderbare ist vielmehr, daß wir mit solchen Maschinen etwas anfangen können. Das Geheimnis liegt darin, daß Zeichen im Computer kalkulierte und kalkulierende Zeichen sind. Diesen Kniff kannte Hannas Freund Faber noch nicht.

In der nüchternen Art, in der Elisabeth Walther über Semiotik spricht, in den formalen und numerischen Schemata, die sie und ihre Mitarbeitenden der Peirceschen Semiotik haben ange-deihen lassen, weht ein vorbereitender Hauch jenes Kalkulierten und Kalkulierenden. Sie selbst wird ohnehin mit den Augen zwinkern.

Anmerkungen

- 1 Ich werde dem ermüdenden Kampf mit der politisch korrekten Form im Deutschen ausweichen.
- 2 Er meint damit, daß eine anständige und voll funktionsfähige Software dadurch uninteressant wird, daß eine andere existiert, die die gleichen Funktionen erfüllt, aber darüber hinaus noch weitere.
- 3 Die unglaubliche Mächtigkeit der informatischen Bemühung zeigt sich in einem Moment, das hier auf-scheint, dem ich aber im Rahmen dieses Textes nicht weiter nachgehen kann. Die Beschreibungsmittel selbst nämlich – da formalisierte, konventionelle, zeichenhafte Gegenstände – unterliegen dem informatischen Blick, d.h. sie werden beschrieben und zu Software gemacht. Die Informatik trägt in sich den ewigen Stachel zur nächsten und wieder nächsten Abstraktion. In diesem Stachel drückt sich am klarsten der semiotische Charakter ihrer Dinge aus: Wir haben es mit einer grandiosen Semiose zu tun, die mehr ist als Weltinterpretation, nämlich Weltveränderung.
- 4 Das Programm, durch einen endlichen Text dargestellt, ist in diesem Sinne die Beschreibung der Steuerung einer Maschine. Rolf Todesco würde sagen: das Programm *ist* die Steuerung.
- 5 Denn nach der Churchschen These berechnen Computer berechenbare Funktionen. Daß die praktische Berechenbarkeit und die Behandlung von vielerlei Fehlerquellen das Bild noch verkomplizieren, ignoriere ich.
- 6 Hier ist nicht der Ort, die Debatte um die Informatik als eine Gestaltungswissenschaft aufzugreifen, wenn-gleich dies gerade in diesem Zusammenhang naheläge und reizvoll wäre. Stellvertretend für die ausgiebige Literatur hierzu s. etwa Winograd & Flores 86, Greenbaum & Kyng 91, Rolf in: Coy et al. 92.
- 7 Dies mag die Stelle sein, an der auf die wichtigsten Arbeiten hinzuweisen ist, die es zu semiotischen Ansät-zen in oder für die Informatik gibt. Ich tue dies ungerechterweise nur summarisch, obwohl das Feld gerade jetzt allmählich als vielversprechend erkannt wird, vgl. Andersen et al 96. Da es meine Absicht mit diesem Text nicht ist, eine umfassende, abgerundete Darstellung zu geben, sei der interessierten Leserin anempfohlen, gelegentlich das eine oder andere der folgenden Werke zu konsultieren: Andersen 90, Andersen et al. 93, Bogarin 89, Figge 91, Gorn 68, Jorna 90, Jorna et al. 93, Maranda 88, Nadin 88, Nake 74, Nake 93, Siek-kenius 93, Zemanek 66.
- 8 In die Dynamik des *Clickens* nämlich, die sog. direkte Manipulation.
- 9 Für die Interaktion ist die semiotische Betrachtung als Koppelung von Zeichen- und Signalprozeß in Nake 94 dargestellt.
- 10 Diese Wörter schließen an Flusser an. Sie wären näher zu erläutern, was hier unterbleiben muß. Der Charak-ter der syntaktisch reduzierten algorithmischen Semiosen wird in der schörten Arbeit von Sieckenius de Sou-za (Sieckenius 93) in anderer Terminologie entwickelt.
- 11 Peirce sagt, wir können nicht anders als in Zeichen denken.

Literatur

- Anderson, Peter Bøgh, *A theory of computer semiotics. Semiotic approaches to construction and assessment of computer systems*; Cambridge University Press, Cambridge 1990.
- Andersen, P. B., Holmqvist, B., Jensen, J. F. (eds.): *The computer as medium*; Cambridge University Press, Cambridge 1993.
- Andersen, P. B., Nadin, M., Nàke, F. (eds.): *Informatics and semiotics*; Dagstuhl-Seminar-Report 135, IBFI Schloß Dagstuhl, Wadern 1996.
- Bogarin, Jorge: *Semiotik der Automaten, Algorithmen und Formalen Sprachen*; Diss. Univ. Stuttgart 1989.
- Bonsiepe, Gui: *Interface. Design neu begreifen*; Bollmann, Mannheim 1996.
- Bürdek, Bernhard E.: *Design. Geschichte, Theorie und Praxis der Produktgestaltung*; DuMont, Köln 1991.
- Coy, W., Nake, F., Pflüger, J. M., Rolf, A., Seetzen, J., Siefkes, D., Stransfeld, R. (Hgg.): *Sichtweisen der Informatik*; Vieweg, Braunschweig 1992.
- Figge, Udo L.: „Computersemiotik“, in: *Z. f. Semiotik* Jg. 13, 1991, S. 321-330.
- Frisch, Max: *Homo faber*; Suhrkamp, Frankfurt/Main 1977 (zuerst 1957).
- Gorn, Saul: „The identification of the computer and information sciences: Their fundamental semiotic concepts and relationships“, in: *Foundations of Language* 4, 1968, S. 339-372.
- Greenbaum, Joan, Kyng, Morten (eds.): *Design at work. Cooperative design of computer systems*; Lawrence Erlbaum, Hillsdale, NJ 1991.
- Hilbert, David: *Grundlagen der Geometrie*; B. G. Teubner, Stuttgart 1962 (nach der fünften Auflage von 1922).
- Jorna, René J.: *Knowledge representation and symbols in the mind*; Stauffenburg, Tübingen 1990.
- Jorna, R. J., Heusden, B. van, Posner, R. (eds.): *Signs, search and communication. Semiotic aspects of artificial intelligence*; de Gruyter, Berlin 1993.
- Maranda, Pierre: „Semiotics and computers: The advent of semiotronics?“, in: T. A. Sebeok, J. Umiker-Sebeok (eds.): *The semiotic web*; Mouton, de Gruyter, Berlin 1988, S. 507-533.
- Marx, Karl: *Das Kapital*. MEW Bd. 23. Dietz, Berlin 1970.
- Mullet, Kevin, Sano, Darrell: *Designing visual interfaces*; Prentice Hall, Hamel Hempstead 1995.
- Nadin, Mihai: „Interface design and evaluation – semiotic implications“, in: H. R. Hartson, D. Hix (eds.): *Advances in human-computer interaction*; Vol. II. Ablex, Norwood, NJ 1988, S. 45-100.
- Nake, Frieder: *Ästhetik als Informationsverarbeitung*; Springer, Wien/New York 1974.
- : „Informatik und die Maschinisierung von Kopfarbeit“, in: Coy et al. 92: 181-201.
- (Hgg.): *Die erträgliche Leichtigkeit der Zeichen. Ästhetik, Semiotik, Informatik*; Agis, Baden-Baden 1993.
- : „Human-computer interaction: signs and signals interfacing“, in: *Languages of Design* Vol. 2, No. 3, Oct. 1994, S. 193-205.
- Schelhowe, Heidi: *Das Medium aus der Maschine*; Campus, Frankfurt/Main, New York 1997.
- Sohn-Rethel, Alfred: *Geistige und körperliche Arbeit*; Suhrkamp, Frankfurt/Main 1972 (rev. u. erg. Neuauflage VCH Verlagsges., Weinheim 1989).
- Sieckenius de Souza, Clarisse: „The semiotic engineering of user interface languages“, in: *Int. J. Man-Machine Studies* 30, 1993, S. 753-773.
- Taylor, Frederick Winslow: *Die Grundsätze wissenschaftlicher Betriebsführung*; neu hg. u. eingeleitet v. Walter Bungard u. Walter Volpert. Beltz, Weinheim 1995.
- Walthert, Elisabeth: *Allgemeine Zeichenlehre*; Deutsche Verlagsanstalt, Stuttgart 1974/1979.
- Winograd, Terry, Flores, Fernando: *Understanding computers and cognition. A new foundation for design*; Ablex, Norwood, NJ 1986 (dt.: *Erkenntnis Maschinen Verstehen*. Rotbuch, Berlin 1989).
- Zemanek, Heinz: „Semiotics and programming languages“, in: *Comm. ACM* 9, 1966, S. 139-143.

Inhalt

Udo Bayer/ Juliane Hansen/ Karl Gfesser	5	Grußwort / Foreword
Ottomar Hartwig	7	Ein Bildzeichen für Elisabeth Walther-Bense zum 75. Geburtstag
Gérard Deledalle	8	Peirce, les Catégories et les Signes
Rosemarie und Fried Alstaedter	23	An Elisabeth
Frieder Nake	24	Der semiotische Charakter der informatischen Gegenstände
Georg Nees	36	Die Blindschleichen, das Eisenerz und die Zeichen. Semiotisch/kybernetische Erinnerungen und Vorahnungen
Wil Frenken	49	Für Elisabeth. PRO CAPTU LECTORIS HABENT SUA FATA LIBELLI
Elisabeth Emter	52	<i>Augenblick</i> . Eine Zeitschrift wider die metaphysische Behaglichkeit
Armin Mehling	60	Geburtstagsgruß
Wojciech H. Kalaga	61	Signification and Objects
Betty Leirner	71	espássaro
Jan Peter Tripp	73	<<Pauline>> (Noch 'ne Blume für E.)
Dinda L. Gorrée	74	Translation: Between Imaging, Modeling, and Manipulation
Angelika Jakob	84	Semiramis der Semiotik
Hans Brög	85	Am Rande der Semiotik
Karel Trinkewitz	91	Bernard Bolzanos Haus in Prag als angeblicher Tatort eines Mordes im Jahr 1848
Dušan I. Bjelić	94	The Levitational Physics of Icons and the Gravitational Theology of Newton
Lee Lichterloh	113	Komposition mit Schwarz
Rudolf Haller	114	Das Fortschreiten der Erkenntnis. Zur Verwendung semiotischer Zusammenhänge durch Benedictus de Spinoza
Frue Cheng	118	Neue Darstellung der Zeichenoperationen
Angelika Karger	128	Zeichenwirkung als philosophische Aufgabe
Jens-Peter Mardersteig	145	Faul im August
Udo Bayer	147	Zur Semiotik der Gartenkunst
M. Drea	165	Le monde en miniature

Karl Herrmann	167	Anwendung semiotischer Vorstellungen zur Erzeugung erkenntnistheoretischer Modelle
Thomas Gil	181	Der Zeichenbegriff in John Lockes empiristischer Erkenntnistheorie
Solange Magalhães	189	S/ Título
Magdolna Orosz	190	"Du kannst nur denken durch den Mittler Sprache." Vermittlung und zeichenhafte Welt in der deutschen Romantik
Reinhard Döhl	203	zuerst wurden die poetiken außer kurs gesetzt - dann kam der reim abhanden - schließlich fehlten sogar die worte. aprèslude
Helmut Kreuzer	209	Hiršals Jugendwelt. Oder eine "ungewohnte Form" der Autobiographie
Almir Mavignier	215	Konvex/Konkave Linie
Ilse Walther-Dulk	216	Auf der Suche nach der Philosophie Marcel Prousts
Xu Hengchun	232	Eine Skizze von Kulturuntersuchung
Vera Molnar	238	Variations Ste.-Victoire 1989-96
Barbara Wichelhaus	244	Der kreative Aufbau von Bedeutungen durch Malen und Zeichnen im Kindesalter
Engelbert Kronthaler	259	Du sollst Dir kein Bild machen ...
Karl Gfesser	274	Vorbemerkungen zu einer semiotischen Textanalyse
Maria Heyer-Loos	297	Montierte Landschaft
Alfred Toth	298	Auf dem Weg zur ersten semiotischen Grammatik
Hariss Kidwaii	311	Die Basistheorie der Semiotik und die Kleine Matrix
Wolfgang Kiwus	318	Computergrafiken
Herbert Heyer	320	Über asymptotisch fehlerfreie Übertragbarkeit von Information
Josef Klein	335	Über Intention und Intension in Ansehung des Aufbaus der deontischen Modalitäten - Zur normsemiotischen Kritik des Extensionalismus
Gerald L. Eberlein/ Angelika Karger	345	Semiotische Analyse eines sozio-kulturellen Phänomens am Beispiel von UFO-Gläubigkeit
Anita Kernwein	355	Bibliographie der Schriften Elisabeth Walthers