

Till A. Heilmann

Worin haust ein Computer? Über Seinsweisen und Gehäuse universaler diskreter Maschinen

2019

<https://doi.org/10.25969/mediarep/3958>

Veröffentlichungsversion / published version

Sammelbandbeitrag / collection article

Empfohlene Zitierung / Suggested Citation:

Heilmann, Till A.: Worin haust ein Computer? Über Seinsweisen und Gehäuse universaler diskreter Maschinen. In: Christina Bartz, Timo Kaerlein, Monique Miggelbrink u.a. (Hg.): *Gehäuse: Mediale Einkapselungen*. Paderborn: Fink 2019 (Schriftenreihe des Graduiertenkollegs "Automatismen"), S. 36–51. DOI: <https://doi.org/10.25969/mediarep/3958>.

Erstmalig hier erschienen / Initial publication here:

<https://nbn-resolving.org/urn:nbn:de:hbz:466:2-33633>

Nutzungsbedingungen:

Dieser Text wird unter einer Creative Commons - Namensnennung 4.0/ Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz finden Sie hier:

<https://creativecommons.org/licenses/by/4.0/>

Terms of use:

This document is made available under a creative commons - Attribution 4.0/ License. For more information see:

<https://creativecommons.org/licenses/by/4.0/>

TILL A. HEILMANN

WORIN HAUST EIN COMPUTER?
ÜBER SEINSWEISEN UND GEHÄUSE
UNIVERSALER DISKRETER MASCHINEN

Dieser Beitrag schlägt eine begriffliche Verschiebung und Erweiterung des Ausdrucks ‚Gehäuse‘ für den Fall universaler diskreter Maschinen¹ vor. These ist, dass die in der Medienwissenschaft weit verbreitete Vorstellung von (Computer-)Gehäusen als verhüllenden ‚Deckelhauben‘, die es in medienarchäologischer Absicht zu öffnen gelte, gedanklich zu kurz greift. Vielmehr sind ‚Gehäuse‘ – heuristisch aufgefasst als vielgestaltige, apparativ wie sozio-kulturell geformte ‚Grenzflächen‘ des operativen Zusammenhangs von Hard- und Software – konstitutiv für die historisch veränderlichen Seinsweisen von Computern, insofern sie nicht nur verschließend, sondern ebenso sehr *erschließend* wirken: Sie machen die jeweiligen Maschinen für ihre Umgebung erst verfügbar. Die Darstellung dreier historischer Wegmarken der Computerentwicklung soll diese These illustrieren und plausibilisieren.

Unter der Deckelhaube

Das Zeitalter des massentauglichen *personal computing* begann mit einem Gehäuse, mit einem Kunststoffgehäuse genauer gesagt. Als Steve Jobs 1976 am Produktdesign des Apple II arbeitete, war sein Ziel ein ‚freundlicher‘ Computer, „ready to run“². Im Gegensatz zum Apple I, der als ‚nackter‘ Platinenrechner ohne Gehäuse (und auch ohne Peripheriegeräte wie Tastatur oder Bildschirm) verkauft wurde und sich an einen kleinen Kreis begeisterter Hobbybastler richtete, sollte Apples zweites Modell ein breites Publikum ansprechen, das einen kompletten, gebrauchsfertigen Computer erwartete. Für das Gehäuse des neuen Rechners ließ sich Jobs im Warenhaus Macy’s von den dort ausgestellten Elektrogeräten inspirieren, wobei es ihm die Cuisinart-Küchenmaschinen besonders angetan hatten.³ Der Apple II sollte schon äußerlich wie ein im täglichen Gebrauch stehendes, leicht zu bedienendes *consumer product* daherkommen, ein „appliance computer“ oder Haushaltscomputer

¹ Vgl. Alan M. Turing, „Rechenmaschinen und Intelligenz“, in: ders., *Intelligence Service. Schriften*, hg. v. Bernhard J. Dotzler und Friedrich Kittler, Berlin, 1987, S. 147-182.

² Vgl. Walter Isaacson, *Steve Jobs: The Exclusive Biography*, New York, NY, 2011, S. 65.

³ Vgl. ebd., S. 67.

eben, wie es in einem Vorabbericht der Zeitschrift *BYTE* hieß.⁴ Dazu gehörte gerade auch die richtige Verkleidung des Rechners. Statt für eine der damals üblichen Umhüllungen aus Blech, die verhältnismäßig einfach und günstig zu fertigen waren und eine gute elektromagnetische Abschirmung der Bauteile boten, entschied sich Jobs für ein aufwendiger herzustellendes, teureres Gehäuse aus Kunststoff, „a beautiful plastic package [...] using some of the nice plastic technology that had evolved over the past years“⁵. Mit seiner ‚warmen‘, professionell anmutenden Plastikverkleidung sollte sich der Apple II möglichst nahtlos in den heimischen Bestand aus Staubsauger, Haartrockner, Bügeleisen usw. einfügen. Und folgerichtig wurde er in einer der ersten großen Werbekampagnen des Unternehmens visuell in eine Reihe mit Elektrogeräten gestellt, an deren dem Computer entgegengesetztem Ende wohl nicht zufällig einer der von Jobs geschätzten Cuisinart-Mixer stand (siehe Abb. 1).



1 – „Introducing Apple II“, *BYTE* 2, 6 (1977), S. 14

⁴ Carl Helmers, „A Nybble on the Apple“, in: *BYTE* 2, 4 (1977), S. 10, online unter: http://archive.org/stream/byte-magazine-1977-04/1977_04_BYTE_02-04_Baudot_Machines, zuletzt aufgerufen am 11.03.2016.

⁵ Steve Jobs/Maggie Canon/Paul Freiberger, „Apple’s Steven P. Jobs Talks to IW (Interview)“, in: *InfoWorld* 4, 9 (1982), S. 12-15: 12, online unter: <https://books.google.de/books?id=gT4EAAAAMBAJ>, zuletzt aufgerufen am 11.03.2016.

Blickt man auf die Geschichte des Personal Computers zurück, mag das Produktdesign des Apple II als symbolischer Beginn einer Entwicklung erscheinen, deren vorläufigen End- und Höhepunkt die digitalen Gadgets von heute markieren: eine Tendenz zunehmender Kapselung von Computertechnik in Gehäusen (gleich welchen Materials) zum Zweck maximierter Käuferakzeptanz und Bedienerfreundlichkeit. Während die frühesten PCs wie der Apple I oder der MITS Altair 8800 von den Nutzerinnen und Nutzern noch selbst zusammengebaut werden mussten und die ersten breitenwirksamen Maschinen wie der Apple II oder der IBM 5150 offene Plattformen darstellten, deren technische Details weitgehend dokumentiert waren und die mit ihren Steckplätzen für Erweiterungskarten zum Aufschrauben der Gehäuse geradezu aufforderten⁶, lassen sich aktuelle Notebooks, Tablets, Smartphones, Smartwatches oder Fitness-Tracker von Laien kaum oder überhaupt nicht öffnen (und also nur mehr schwerlich reparieren oder gar erweitern).

So gesehen wirkt die Entwicklungsgeschichte persönlicher Computertechnik wie eine Verfallsgeschichte, die von begeisterten Bastlern der ersten Stunde über kundige Käufer von IBM-kompatiblen Schreibtischmodellen zu entmündigten Endbenutzern von iPhones, Kindles und Fitbits führt. Und es ist ja auch nicht zu bestreiten: Setzten frühe Computerbausätze fundierte Elektronikkenntnisse und Lötterfahrung voraus⁷, und wurde PC-Besitzern in den 1980er-Jahren immerhin noch zugetraut, Jumper an Platinen stecken und Zusatzkarten selbst einbauen zu können, gehört es mittlerweile zur Produktphilosophie von Apple und anderen Firmen, Geräte ganz ohne Bedienungsanleitungen zu verkaufen. Ihr Gebrauch soll selbsterklärend bzw. ‚intuitiv‘ sein – wobei sich die intendierte Selbstverständlichkeit der Bedienung eben nicht zuletzt am Gehäuse der Geräte äußert. (Man denke nur an den ikonischen einzelnen Home-Button des iPhone oder Steve Jobs' hartnäckige Weigerung, eine Computermaus mit mehr als einer Taste anzubieten.) Die Gehäuse aktueller Digitalgeräte schützen also nicht nur die einzelnen Komponenten der Geräte vor „äußeren Einflüssen (wie Schmutz, Wasser, Elektrizität, Druck etc.)“⁸, sie schützen ebenso ihre Nutzerinnen und Nutzer vor der Zumutung, sich mit diesen Komponenten überhaupt auf theoretischer wie praktischer Ebene auseinandersetzen zu müssen. Vom Apple II bis zum Samsung Galaxy S7 manifestiert sich in den Gehäusen der Geräte die Idee einer Domestizierung von Computertechnik, deren Ziel das vollständige Zurücktreten der Funktionsweise

⁶ Siehe etwa zum IBM 5150 das *IBM Personal Computer Technical Reference Manual*, Boca Raton, FL, 1981, online unter: <http://www.pcjs.org/pubs/pc/reference/ibm/5150/techref/>, zuletzt aufgerufen am 11.03.2016.

⁷ Überdies mussten Maschinen wie der MITS Altair 8800 anfangs selbst programmiert werden, da es die entsprechende Softwareindustrie noch nicht gab; vgl. Steven Levy, *Hackers: Heroes of the Computer Revolution*, Garden City, NY, 1984, S. 203-206 und S. 224-231.

⁸ „Computergehäuse“, online unter: <https://de.wikipedia.org/wiki/Computergehäuse>, zuletzt aufgerufen am 02.02.2016.

universaler diskreter Maschinen hinter ihr reibungsloses Funktionieren im alltäglichen Einsatz ist.

Die hier umrissene historische Einschätzung – die offenkundig an einer ziemlich umgangssprachlichen Auffassung des Ausdrucks ‚Gehäuse‘ hängt – gehört zu den Gemeinplätzen des medienwissenschaftlichen Nachdenkens über Computer, zumindest einer besonders wirkmächtigen Richtung desselben. Diese Denkrichtung, heute meist als Medienarchäologie bezeichnet, wurde im deutschsprachigen Raum von Friedrich Kittler begründet, der ihr auch den wohl prominentesten Ausdruck verliehen hat. Gemäß Kittler stellen Gehäuse, kurz gesagt, Verhüllungen dar: Hüllen, die den Blick auf die Konstruktion und Funktionsweise technischer Apparate versperren und so deren rechtes Verständnis verunmöglichen. Vordringliche Aufgabe der Medienwissenschaft muss es nach Kittler daher sein, die untersuchten Medien(apparate) ihrer Gehäuse zu entkleiden und ihr ‚Innenleben‘ zu studieren.⁹ Andernfalls bekomme man „immer nur jene Schauseite zu fassen, die die Elektronikindustrie bewusst so auslegt, dass die Innereien der Apparate unter ihrer Deckelhaube bleiben“¹⁰. Gegen die Absicht der Industrie gelte es, die Deckelhauben der Geräte zu öffnen, um, wie Kittler sich ausdrückt, „im Synthesizersound der Compact Disc den Schaltplan selber zu hören oder im Lasergewitter der Diskotheken den Schaltplan selber zu sehen“¹¹.

Der medientechnische Entlarvungsgestus, der sich in den zitierten Stellen bekundet, ist insofern bemerkenswert, als er Kittlers antihermeneutischem Programm der „Austreibung des Geistes“¹² zuwiderläuft. Kittlers technologische Suche nach dem Medium unter der Deckelhaube des Apparates wiederholt ja in gewisser Weise nur das Muster der hermeneutischen Suche nach dem Sinn unter der ‚Oberfläche‘ von Texten.¹³ Vor allem aber ist die sich gleichermaßen auf- wie abgeklärt gebende Haltung Kittlers aus mehreren Gründen problematisch, wenn es um universale diskrete Maschinen geht. Zunächst, was sieht man, wenn man das Gehäuse eines Computers öffnet? Man sieht zugleich sehr viel und sehr wenig: viel, weil Computer noch im Falle handlicher Geräte wie aktueller Smartphones aus Dutzenden von Bauteilen bestehen¹⁴; wenig indes, weil einem der Anblick dieser Bauteile kaum etwas darüber verrät, wie sie arbeiten. Vor der nanoskopischen Dimension und der strukturellen

⁹ Als jüngerer Beispiel für diesen Ansatz siehe „Prof. Dr. Wolfgang Ernst erläutert Sinn und Zweck des medienarchäologischen Fundus“, 27.01.2010, 00:04:23-00:06:04, online unter: <https://www.youtube.com/watch?v=Jq1jkkPqXM8>, zuletzt aufgerufen am 17.02.2016.

¹⁰ Friedrich Kittler, *Optische Medien. Berliner Vorlesung 1999*, Berlin, 2002, S. 24 f.

¹¹ Friedrich Kittler, *Grammophon, Film, Typewriter*, Berlin, 1986, S. 5.

¹² Siehe Friedrich Kittler (Hg.), *Austreibung des Geistes aus den Geisteswissenschaften. Programme des Poststrukturalismus*, Paderborn, 1980.

¹³ Vgl. Till A. Heilmann, „Die Oberflächlichkeit des Digitalen“, in: Christina Lechtermann/Stefan Rieger (Hg.), *Das Wissen der Oberfläche. Epistemologie des Horizontalen und Strategien der Benachbarung*, Zürich, Berlin, 2015, S. 253-266: 257 f.

¹⁴ Siehe für ein extremes Beispiel „Apple Pencil Teardown“, online unter: <https://www.ifixit.com/Teardown/Apple+Pencil+Teardown/52955>, zuletzt aufgerufen am 27.02.2016.

Komplexität elektronischer Komponenten versagt nicht nur bloßer Augenschein. Auch eine optische Vergrößerung etwa der integrierten Schaltkreise würde nicht weiterhelfen. Schon ein vergleichsweise einfach aufgebauter Mikroprozessor wie der inzwischen über vierzig Jahre alte Intel 4004 übersteigt die Grenzen des visuellen Nachvollzugs seiner Funktionsweise.¹⁵

Es ist deshalb nicht verwunderlich, dass Kittler seine Kritik der Computerindustrie vornehmlich als eine Kritik der Software formuliert hat. Die Deckelhauben, unter die bei Computern zu schauen er forderte, das sind für ihn die verschiedenen Codeschichten, die auf dem Befehlssatz des Prozessors aufrufen und den Usern eine vollständige Kontrolle der Hardware-Ressourcen des Systems verwehren: höhere Programmiersprachen, „Protected Mode“-Betriebssysteme, grafische Benutzeroberflächen und nicht-quelloffene Anwendungsprogramme.¹⁶ Als Gegenstrategie zur bedienerfreundlichen, aber erkenntnisfeindlichen Abstraktion so verstandener Software von den basalen Funktionsmechanismen der Hardware propagierte Kittler einen selbstbestimmten Umgang mit Computern, d. h. das Programmieren in Maschinensprache (oder wenigstens einer vergleichsweise maschinennahen Sprache wie C). Damit käme man, so sein Versprechen, dem ‚Kern‘ von Computerhardware, der „mathematischen Transparenz“ eines idealen Prozessors, möglichst nahe.¹⁷

Dieses Versprechen führt jedoch wieder auf die Probleme mit der Kittler'schen ‚Entlarvung‘ von Computertechnik zurück. Denn was nach der Reduktion um Software von einem Computer übrig bleibt, ist eben bloße Hardware.¹⁸ Und die interessiert, wenn es um die spezifische mediale Leistung von Computern geht, wenig. Zwar unterscheiden sich einzelne Maschinen hinsichtlich ihrer Bauweise¹⁹ und ihrer Rechenkapazitäten erheblich voneinander (und die Unterschiede können, gerade für die Verarbeitung von Bild-, Ton- und Videoformaten, entscheidend sein.) Was ihre theoretische Leistungsfähigkeit angeht, dürfen nach Alan Turing aber alle universalen diskreten Maschinen als gleich mächtig gelten.²⁰ Nicht nur lässt sich also die Funktionsweise eines Computers durch das Öffnen des Gehäuses kaum klären. Zur Beurteilung seiner medialen Fähigkeiten ist dies im Prinzip auch gar nicht nötig: Computer sind diejenigen Maschinen, die theoretisch das Funktionieren aller

¹⁵ Siehe den Schaltplan des Chips, online unter: <http://www.4004.com/mcs4-masks-schematics-sim.html>, zuletzt aufgerufen am 27.02.2016.

¹⁶ Siehe Friedrich Kittler, „Protected Mode“, in: ders., *Draculas Vermächtnis. Technische Schriften*, Leipzig, 1993, S. 208-224; ders., „Es gibt keine Software“, in: ders., *Draculas Vermächtnis. Technische Schriften*, Leipzig, 1993, S. 225-242; ders., „Hardware, das unbekannte Wesen“, in: Sybille Krämer (Hg.), *Medien, Computer, Realität*, Frankfurt/M., 1998, S. 119-132.

¹⁷ Vgl. Kittler (1993), *Protected Mode*, S. 223.

¹⁸ Zu Kittlers Phantasma ‚reiner Hardware‘ siehe Kittler (1993), *Es gibt keine Software*, S. 241 f.

¹⁹ Zumindest im PC-Bereich ist die Prozessorarchitektur jedoch durch die vorherrschende IA-32-Architektur von Intel genormt.

²⁰ Vgl. Alan M. Turing, „Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem“, in: ders., *Intelligence Service. Schriften*, hg. v. Bernhard J. Dotzler und Friedrich Kittler, Berlin, 1987, S. 17-60.

anderen Maschinen zu simulieren vermögen. Solche Simulationen setzen geeignete Hardware wohl voraus, realisieren sich aber erst im Vollzug durch Software. Was man an Hardware unter einem Gehäuse zu sehen bekommt, existiert nicht in derselben Weise wie die medialen Prozesse, die darauf laufen. Als arbeitende Maschine „*ek-sistiert*“ ein Computer, mit Christoph Tholen gesprochen, „nur in seiner vielgestaltigen Metaphorizität“ wechselnder Programme, die ihn mal zur Schreibmaschine machen, mal zum Fotolabor, mal zum Telefon usw.²¹ Es gibt kein Wesen eines Computers, das mit den ‚Innereien‘ des Apparates gleichzusetzen und also unter dessen Deckelhaube zu finden wäre.

Die Frage danach, was im Gehäuse eines Computers steckt, führt in die Irre, will man etwas über dessen mediales Wirken erfahren. Und sie tut dies nicht zuletzt deshalb, weil sie von der umgangssprachlichen Bedeutung des Ausdrucks ‚Gehäuse‘ ausgeht und suggeriert, ein (vorgeblich wesentliches) Inneres des Computers, das häufig mit dem Prozessor identifiziert wird²², von einem (unwesentlichen) Äußeren scheiden zu können. Nicht allein die jeweils laufenden Programme aber machen zusammen mit dem Prozessor eines Computers ein arbeitendes System aus. Es gehört dazu noch eine technische Umgebung, in welcher der Prozessor haust: Neben der geeigneten Stromversorgung können das Komponenten wie Arbeitsspeicher, Netzwerkadapter und Grafikkarten sein, Schnittstellengeräte zur Ein- und Ausgabe von Daten sowie angeschlossene Systeme. Die so verstandene technische Umgebung mag ein Desktop-PC mit Bildschirm und Tastatur sein, ein Smartphone, eine Digitalkamera oder ein Internetrouter, aber auch eine Waschmaschine, eine Interkontinentalrakete oder ein PKW.²³

Gute Gründe sprechen also dafür, den Ausdruck ‚Gehäuse‘ in medienwissenschaftlichen Untersuchungen von Computern anders aufzufassen, als er gemeinhin gebraucht wird. Statt bloß für die materielle Umhüllung eines Geräts (wie in der eingangs geschilderten Episode um den Apple II) oder schon gar für die absichtliche Verhüllung einer vermeintlichen komputationellen Essenz (wie bei Friedrich Kittler) könnte das Wort ‚Gehäuse‘ in heuristischer Absicht und in einem strategisch weiten Sinne verwendet werden. Es würde dann, in einer ersten begrifflichen Annäherung, ganz allgemein für die ‚Grenzflächen‘ stehen, die ein gegebener Computer (als operativer Zusammenhang von Hardware- und Software-Komponenten) gegenüber der Umwelt aus Körpern und Dingen aufweist. Diese ‚Grenzflächen‘ wären dabei nicht nur hinsichtlich ihrer schützenden, d. h. den Computer von der Umwelt ab- oder

²¹ Georg Christoph Tholen, *Die Zäsur der Medien*, Frankfurt/M., 2002, S. 52.

²² In der Halbleiterindustrie wird die schützende Ummantelung eines ‚nackten‘ Mikrochips in Kunststoff, Keramik oder Metall samt der Anschlussstellen tatsächlich ‚Gehäuse‘ oder (englisch) *package* genannt.

²³ Rund 98 % aller Mikrochips werden in sogenannten eingebetteten Systemen verbaut, nicht in Desktop-Rechnern oder Notebooks, die oftmals *pars pro toto* mit Computern überhaupt gleichgesetzt werden.

verschließenden, Eigenschaften, sondern ebenso hinsichtlich ihrer sozusagen überbrückenden, d. h. den Computer an die Umwelt *an-* und ihn jener *erschließenden*, Qualitäten zu denken.²⁴ Und sie dürften nicht ausschließlich als materiell-technische Formen begriffen werden, sondern müssten auch soziale, ökonomische, kulturelle, rechtliche u. a. Ausformungen der technischen Vermittlungsleistung von Computern umfassen. Die Frage nach den ‚Gehäusen‘ von Computern betreffe damit die geschichtlich wandelbaren Seinsweisen universaler diskreter Maschinen selbst, nicht bloß deren wechselnde Erscheinungsbilder.

Rückblick: Rechnerraum, Kommandozeile, Formfaktor

Bevor Mikroprozessoren auf Halbleiterbasis kleine und günstige Geräte wie PCs möglich machten, hatten universale diskrete Maschinen eine heutigen Erfahrungen von aktueller Computertechnik eher unvertraute Gestalt. In den 1950er- und 1960er-Jahren wurden sie – den Fall militärischer Spezialgeräte wie immer ausgenommen – hauptsächlich als sogenannte Mainframes realisiert: Großrechner für wissenschaftliche und betriebswirtschaftliche Anwendungen, teuer in der Anschaffung und im Unterhalt, wartungsintensiv, schwierig zu programmieren und umständlich zu bedienen.²⁵ Und dem Stand der Technik geschuldet waren die Großrechner zuerst einmal genau das: groß. Sie bestanden typischerweise aus mehreren einzelnen, freistehenden Komponenten, die miteinander verkabelt wurden. Abbildung 2 zeigt ein damals wohl typisches System aus den 1950er-Jahren, eine IBM 701-Anlage am US-amerikanischen Kernwaffenforschungszentrum Lawrence Livermore National Laboratory. In der Mitte ist das Kontrollpult des Computers zu sehen, links daneben ein Kartenleser; hinter dem Kontrollpult steht die Recheneinheit und dahinter zwei Magnetkernspeicher, links davon die Stromversorgung; am rechten Bildrand ist ein Kathodenstrahlröhrenbildschirm zu sehen und davor ein Zeilendrucker, am linken Bildrand zwei Magnetbandlaufwerke. Zu erkennen sind außerdem Hauben zur Lüftung der Recheneinheit und der Kernspeicher sowie die in den Doppelboden geführten Kabel der Recheneinheit.

²⁴ Der Ausdruck ‚Gehäuse‘ würde so in die Nähe von neueren Interface-Konzepten rücken, die Mensch-Maschine-Schnittstellen weniger gegenständlich-apparativ als vielmehr prozessual-relational verstehen; siehe dazu Sabine Wirth, „Between Interactivity, Control, and ‚Everydayness‘ – Towards a Theory of User Interfaces“, in: Florian Hadler/Joachim Haupt (Hg.), *Interface Critique*, Berlin, 2016.

²⁵ Siehe zu Mainframes Paul E. Ceruzzi, *A History of Modern Computing*, 2. Aufl., Cambridge, MA, 2003, S. 71-74.



2 – IBM 701-Anlage im Lawrence Livermore National Laboratory

Schon aufgrund der verteilten Anlage ist es kaum möglich, hier ‚das‘ Gehäuse des Computers mit einer der Umhüllungen der genannten Komponenten zu identifizieren. Es liegt stattdessen nahe, in diesem Fall den Raum, in dem die Anlage steht, als das ‚Gehäuse‘ des Computers anzusehen. Der Rechnerraum stellte Mainframes seit den 1950er-Jahren die nötige bauliche Infrastruktur und bot den teuren und wartungsintensiven Maschinen durch seine Abgeschlossenheit zudem ein Mindestmaß an Schutz vor äußeren Einwirkungen. Großrechner waren aber nicht nur wegen ihren schieren Dimensionen und den Anforderungen an die Klimatisierung und Stromversorgung in einem eigenen Raum untergebracht. Es war dies auch Ausdruck ökonomischer bzw. arbeitsorganisatorischer Rahmenbedingungen: Die Mainframes konnten nur ein Programm zur selben Zeit ausführen, und ihre Rechenzeit war so kostbar, dass die einzig wirtschaftlich vertretbare Art ihrer Nutzung die Stapelverarbeitung war, das *batch processing*.²⁶ Die Computer wurden also nicht – wie man das heute von PCs, Tablets und Smartphones gewohnt ist – interaktiv bedient und gesteuert (was bedeutet hätte, dass die Maschine einen beträchtlichen Teil der Betriebszeit damit zubringt, auf Eingaben der Benutzerin zu warten). Sie ver-

²⁶ Für eine literarische Bearbeitung des *batch processing* siehe Emil Zopf, *Jede Minute kostet 33 Franken*, Zürich, 1977.

richteten ihre Arbeit weitgehend automatisiert und möglichst unterbrechungsfrei, indem alle zu prozessierenden Programme und Daten vom Systemoperator und Hilfskräften nacheinander (aber mit unterschiedlichen Prioritäten) von Lochkarten auf Magnetbänder überspielt und diese sequenziell ausgelesen bzw. ausgeführt und verarbeitet wurden. Die Ergebnisse wurden dann wieder auf Magnetband zurückgeschrieben und in Lochkarten gestanzt oder auf Endlospapier ausgedruckt.

Um den reibungslosen Ablauf des *batch processing* sicherzustellen, war der Zugang zum Rechnerraum streng kontrolliert und eingeschränkt. Gewöhnliche Nutzerinnen und Nutzer der Anlage (Wissenschaftler, Buchhalter, aber auch Programmierer) durften den Raum nicht betreten und bekamen den Computer in der Regel gar nicht zu Gesicht. Das ‚Gehäuse‘ Rechnerraum erfüllte somit tatsächlich eine verschließende Funktion. Die User mussten ihre auszuführenden Aufträge, in Lochkarten gestanzt, in einem dem Rechnerraum vorgelagerten Büro oder an einem dafür eingerichteten Schalter abgeben. Nach einigen Stunden (manchmal auch am nächsten oder übernächsten Tag) durften sie die Ergebnisse der Ausführung dann, von einem Zeilendrucker auf Endlospapier herausgeschrieben, abholen. An diesem Schema zeigt sich nun die zugleich erschließende Funktion des ‚Gehäuses‘: Da es aus ökonomischen (wie aus technischen) Gründen nicht praktikabel gewesen wäre, die Nutzerinnen und Nutzer selbst an den Computer zu lassen, fungierte der Rechnerraum – in architektonischer wie in organisatorischer Hinsicht – als regulatives Prinzip, das die kostbare Rechenzeit einzelnen Personen überhaupt erst verfügbar machte.

Wie leicht einzusehen ist, gestaltete sich das Programmieren von Computern im Stapelverarbeitungsbetrieb einer Anlage umständlich und zeitraubend. Zwischen der Ausformulierung eines Programms und seiner Ausführung lagen Stunden, und mit der wachsenden Komplexität und den steigenden Anforderungen an die Software nahm zwangsläufig auch die Zahl der Fehler im Code zu. Weil diese aber häufig erst nach einer (abgebrochenen) Programmausführung entdeckt und korrigiert wurden, geriet die Stapelverarbeitung zum Hemmschuh für die Softwareentwicklung.²⁷ Das Problem der zu optimierenden Ressourcenauslastung von Computern wandelte sich mehr und mehr zum Problem ihrer zu optimierenden Programmierung und Steuerung. Bereits am Ende der 1950er-Jahre zeichnete sich daher eine Abkehr vom *batch processing* ab. Weil die Computerhardware durch Innovationen wie Transistoren und Festplatten immer leistungsfähiger, zuverlässiger und günstiger geworden war, schien es nicht mehr bloß denkbar, sondern machbar, ein System mehreren

²⁷ Für eine humoristische Darstellung der Schwierigkeiten mit Lochkarten-Stapelverarbeitung siehe den Kurzfilm *Ellis D. Kropotchev Silent Film* von Arthur Eisenson und Gary Feldman aus dem Jahr 1967, online unter: <http://www.computerhistory.org/revolution/punched-cards/2/211/2253>, zuletzt aufgerufen am 20.07.2016.

Personen gleichzeitig zur Verfügung zu stellen – und so *interactive computing* zu betreiben.²⁸

Die entsprechenden technischen Verfahren – zusammengefasst Timesharing genannt – wurden seit den frühen 1960er-Jahren vor allem an US-amerikanischen Universitäten erprobt und zur Reife gebracht. An Systemen wie dem CTSS (MIT) oder dem DTSS (Dartmouth College) konnten Nutzerinnen und Nutzer über entsprechende Terminals selbst mit dem Computer in Verbindung treten und im laufenden Betrieb beispielsweise Programme schreiben, testen und überarbeiten.²⁹ An umfunktionierten elektrischen Schreibmaschinen und Fernschreibern, die (ggf. über Telefonleitungen) an die jeweilige Rechneranlage angeschlossen wurden, interagierte man mittels alphanumerisch codierter Botschaften mit dem System. Der Technik und Medialität der Schreibmaschinentastaturen und -druckwerke gemäß hatte die Interaktion mit dem Computer also die Form eines zeilenbasierten schriftlichen Dialogs. Vermittelt wurde dieser Dialog durch ein eigenes Programm, den Kommandozeileninterpreter, der im Englischen nicht umsonst *shell* – ‚Schale‘, ‚Hülle‘ oder eben ‚Gehäuse‘ – heißt. Er überwachte die Eingabe von Zeichen auf dem Terminal, analysierte und interpretierte die eingegebenen Zeichenfolgen als Ketten von Befehlen und Parametern, rief entsprechende weitere Programme auf, nahm deren Resultate entgegen und schickte sie als Zeichenfolge wieder an das Terminal zurück.

Weil Computer im Time-Sharing-Betrieb nicht einen Auftrag nach dem anderen erledigten und dem jeweils laufenden Programm somit nicht alle Ressourcen exklusiv überantworten konnten, sondern quasi gleichzeitig die Eingaben verschiedener Nutzerinnen und Nutzer abfragen und deren Prozesse und Daten nebeneinander verarbeiten mussten, war es notwendig, die Verwaltung des Speichers, der laufenden Prozesse, der Lese- und Schreibzugriffe auf Dateien usw. einem Betriebssystem zu überantworten, das sich zwischen die Nutzerinnen und Nutzer und die Hardware schob. Insofern verschloss das ‚Gehäuse‘ des Kommandozeileninterpreters von Timesharing-Systemen die Computer tatsächlich ein Stück weit vor ihren Usern, weil für jeden Einzelnen nur noch ein gewisser Teil der Rechenzeit, des Arbeitsspeichers und der Massenspeicher reserviert war. Der Protected Mode der Intel-Chips, der später Kittlers medientheoretischen Zorn erregen sollte³⁰, kann als Übertragung des Time-Sharing-Prinzips in die Architektur von Mikroprozessoren selbst verstanden werden. Wie geschildert diente diese Betriebslogik jedoch nicht dazu, die Hardware der Maschinen zu verbergen, und schon gar nicht führte sie dazu, dass Benutzer vom Computer ferngehalten wurden. Im Gegenteil: Erst mit Multiuser-Systemen konnten in den 1960er-Jahren die wenigen damals

²⁸ Der einschlägige Text hierzu ist Joseph C. R. Licklider, „Man-Computer Symbiosis“, in: *IRE Transactions on Human Factors in Electronics*, 1 (1960), S. 4-11.

²⁹ Zur Frühgeschichte des Time-Sharing siehe insbesondere die beiden diesbezüglichen Nummern der *IEEE Annals of the History of Computing*, 14, 1-2 (1992).

³⁰ Siehe Kittler (1993), Protected Mode.

vorhandenen Computer überhaupt für interaktive Programmierung und Bedienung erschlossen werden (dank geeigneten Installationen wie dem DTSS mit BASIC-Interpreter sogar Computerlaien; siehe Abb. 3); und erst auf Intels Protected Mode aufsetzende Multitasking-Betriebssysteme ermöglichten es PC-Besitzern in den 1980er-Jahren, sicher mit mehreren Programmen ‚gleichzeitig‘ zu arbeiten. Gegen Kittlers einseitige Behauptung einer wissenspolitischen und technischen ‚Unterwerfung‘ des Subjekts unter Industrieinteressen und einer gezielten Produktion ‚unmündiger User‘ durch den Protected Mode³¹ ist deshalb, in geradezu dialektischer Manier, auf der wechselseitigen Bedingtheit von Entmündigung und Ermächtigung durch das ‚Gehäuse‘ von Multiuser- bzw. Multitasking-Computern zu bestehen.



3 – Studenten des Dartmouth College an DTSS-Terminals;
Foto von Adrian N. Bouchard

Während die Arbeit auf der *shell* einer modernen Terminalemulation heute für kryptisches Expertenwissen steht und als Erkennungsmarke sogenannter *power user* gilt, bildeten Terminal und Kommandozeileninterpreter bis zum Aufkommen des PC am Ende der 1970er-Jahre das gängige ‚Gehäuse‘ interaktiver Computersysteme, mit welchen man als User in Kontakt treten konnte. Über die ‚Grenzflächen‘ aus Schreibmaschinentastaturen und Endlospapier hinaus gelangten Nicht-Spezialisten ab den späten 1970er-Jahren, als Personal Computer die Hard- und Software universaler diskreter Maschinen im kleinen

³¹ Siehe ebd., S. 208.

Maßstab und für vergleichsweise wenig Geld verfügbar machten. Und der durchschlagende Erfolg, den die PC-Industrie im Allgemeinen sowie die IBM-kompatible PC-Plattform im Besonderen ab der Mitte der 1980er-Jahre feierte, verdankt sich zu einem nicht geringen Teil der Idee offener Systeme. Wie eingangs erwähnt, waren die technischen Details vieler früherer Maschinen – auch wenn diese nicht, wie der Altair 8800, selbst zusammengebaut werden mussten – offengelegt. Teils war das, wie im Falle des von Steven Wozniak konzipierten Apple II, auf die Hackerethik der *counterculture* zurückzuführen³²; teils entsprang es aber auch technischen Zwängen und ökonomischen Überlegungen, wie im Fall des ersten IBM-PC, dem IBM Personal Computer 5150: Erstens kaufte IBM, um die Entwicklungszeit möglichst kurz zu halten und das Produkt schnell auf den Markt bringen zu können, wesentliche Teile (außer der vermeintlich entscheidenden Firmware) für die neue Maschine bei Drittherstellern wie Intel und Microsoft ein. Zweitens dokumentierte IBM die Konstruktion der Maschine in allen Einzelheiten³³, damit Drittanbieter zusätzliche Hard- und Software für das System entwickeln konnten. IBM glaubte, den eigenen Rechner mit der Offenlegung seiner Konstruktion möglichst attraktiv zu machen und für dessen schnelle Akzeptanz und Verbreitung zu sorgen, durch das proprietäre, rechtlich geschützte BIOS³⁴ gleichzeitig aber das technische Monopol über das System zu wahren. Dieser Glaube stellte sich jedoch schnell als fataler Fehler für IBMs Geschäftsstrategie wie auch als unerwarteter Glücksfall für die PC-Plattform insgesamt heraus. Denn nach kurzer Zeit bauten Firmen wie Compaq die Funktionen von IBMs BIOS auf legale Weise nach und brachten mit frei erhältlichen Komponenten wie Intel- und AMD-Prozessoren und dem in Lizenz erhältlichen MS-DOS von Microsoft eigene PCs auf den Markt, die kompatibel zu den IBM-Geräten waren, dabei aber günstiger und teilweise auch leistungsfähiger als die ‚Originale‘.³⁵ So wurde

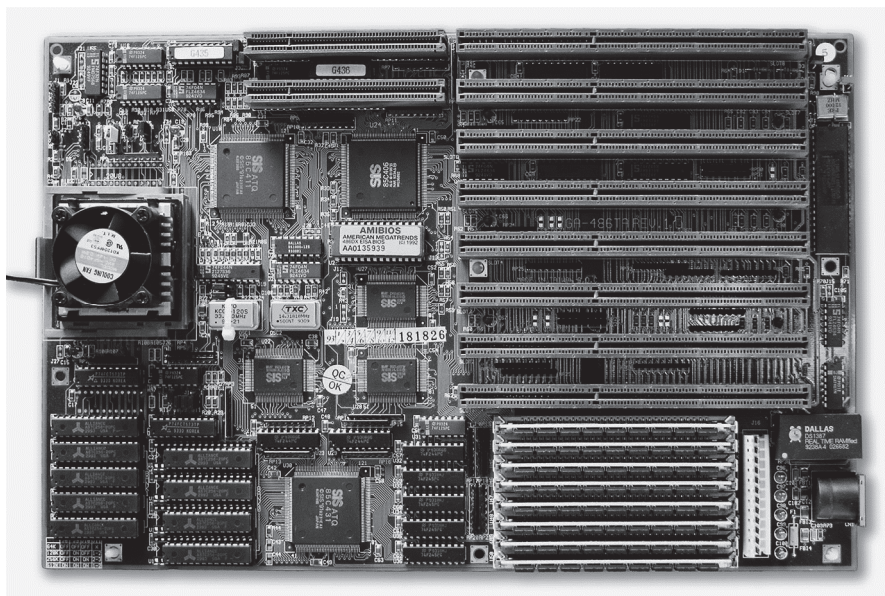
³² Siehe als exemplarischen Fall Lee Felsenstein, „The Tom Swift Terminal or, a Convivial Cybernetic Device“, 1976, online unter: http://www.leefelsenstein.com/wp-content/uploads/2013/01/TST_scan_150.pdf, zuletzt aufgerufen am 11.03.2016; vgl. Michael Swaine/Paul Freiberger, „Lee Felsenstein: Populist Engineer“, in: *InfoWorld* 5, 45 (1983), S. 105-107, online unter: <https://books.google.de/books?id=0C8EAAAAMBAJ>, zuletzt aufgerufen am 11.03.2016; Fred Turner, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*, Chicago, IL, 2006, S. 115.

³³ Siehe *IBM Personal Computer Technical Reference Manual*, Boca Raton, FL, 1981, online unter: <http://www.pcjs.org/pubs/pc/reference/ibm/5150/techref/>, zuletzt aufgerufen am 11.03.2016.

³⁴ Das *Basic Input/Output System* (BIOS) ist die Firmware älterer IBM-kompatibler PCs, die nach dem Einschalten des Computers die Hardware testet und initialisiert und das Betriebssystem startet. Weil das BIOS in der Frühzeit der IBM-PCs die einzelnen Teile der Hardware erst für Betriebssysteme und Programme ansprechbar machte, fungierte der von IBM in der technischen Referenz veröffentlichte und unter Copyright stehende Code des BIOS zugleich als ‚Kopierschutz‘ der Plattform.

³⁵ Für eine unterhaltsame Kurzdarstellung dieser Episode der Computergeschichte siehe Robert X. Cringely, *Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date*, 2. Aufl., New York, NY, 1996, S. 137 und S. 169-174.

die IBM-Plattform Mitte der 1980er-Jahre – ganz im Sinne der Firma, aber ohne dass sie davon noch finanziell profitieren konnte – zum De-facto-Standard der PC-Industrie.



4 – Gigabyte GA-486TA Baby AT Hauptplatine; Foto von Andrew Dunn

Mit dem *reverse engineering* des BIOS alleine war es natürlich nicht getan – auch wenn das fraglos einen der aufwendigsten und teuersten Aspekte des ‚Klonens‘ der IBM-Maschinen darstellte. Compaq und andere Firmen mussten IBMs System ja von Grund auf Stück für Stück nachbauen und dabei sicher gehen, dass die Einzelteile ihrer PCs (vom Netzteil über den Prozessor bis zur Festplatte) reibungslos zusammenspielten. Gefragt waren deshalb industrieweite Normen für PC-Komponenten. Und weil zu Beginn eine einhundertprozentige technische Kompatibilität mit dem System des Marktführers IBM das Ziel von Konkurrenten wie Compaq war, orientierte man sich pragmatisch an der Architektur der entsprechenden Maschinen. Zur Referenz für die Branche wurde die Bauweise der zweiten Generation des IBM-PC: des IBM 5170 oder IBM Personal Computer/AT, kurz PC/AT.³⁶ Der wichtigste Bezugspunkt dieses Computers war seine Hauptplatine, und zwar in der kleineren Baby-Ausführung aus dem Jahr 1985. Ihre Eckdaten gaben vor, was in der Industrie über Jahre hinweg als sogenannter AT-Formfaktor Bestand hatte. Dabei kam

³⁶ Der PC/AT war übrigens just diejenige Maschine, in welcher IBM erstmals den 80286-Prozessor von Intel mit der von Friedrich Kittler geschmähten „Protected Mode“-Funktion verbaut.

es weniger auf den verwendeten 286er-Mikroprozessor an³⁷, sondern auf die Abmessungen der Platine, die Anzahl und Position der Befestigungslöcher, das Layout der Chips und Schnittstellen, die Steckplätze für Erweiterungskarten, die Stromversorgung und das Tastaturinterface. Und weil mit der Größe der Hauptplatine und der Anordnung der darauf befindlichen Steckplätze für Erweiterungskarten, dem Stromanschluss usw. (siehe Abb. 4) zugleich die Hardware-Gestalt eines PCs in entscheidenden Punkten und bis hin zu den Dimensionen des umgebenden Tower-Gehäuses festgelegt war, darf der AT-Formfaktor als ‚Gehäuse‘ der PC-Plattform gelten: Er fixierte, wie die Elemente eines PCs als System zusammen ‚hausten‘; und er machte das System in seinen Elementen dank genauer Spezifikationen als ‚Grenzfläche‘ für ökonomische Kalküle wie für private Basteleien verfügbar.

Bis zum Ende der 1990er-Jahre entsprach die überwiegende Mehrheit aller weltweit gebauten IBM-kompatiblen Desktop-Rechner dem AT-Formfaktor.³⁸ Das auf dem Referenzdesign von IBM beruhende Normenbündel ermöglichte das Entstehen einer globalen PC-Industrie, die über ein Jahrzehnt hinweg (nämlich bis zur Veröffentlichung des nachfolgenden ATX-Formfaktors durch Intel 1995) eine Vielzahl untereinander kombinierbarer und austauschbarer Einzelteile entwickelte und daraus Millionen miteinander kompatibler PCs konstruierte, für welche eine parallel dazu entstandene Softwareindustrie mit Firmen wie Microsoft und Adobe dann immer neue Betriebssysteme, Anwendungsprogramme, Spiele usw. zur Verfügung stellen konnte. Der AT-Formfaktor markierte die Geburt des modernen Personal Computers, d. h. des komponentenbasierten PCs, und seine verschiedenen Nachfolger sorgen bis heute für einheitliche Vorgaben der PC-Architektur, nach denen sich Anbieter mit kommerziellen Geräten wie auch Privatleute beim Zusammenstellen individualisierter Maschinen richten. Im ‚Gehäuse‘ der Formfaktoren setzen sich universale diskrete Maschinen generationenweise als Desktop-Computer fort.

Ausblick: die Welt als/im Gehäuse

Drei Jahrzehnte lang, von den späten 1970ern bis zum Ende der Nullerjahre, wurde die gesamtgesellschaftliche Verbreitung von Computertechnik durch den PC getragen. Aber spätestens mit dem Siegeszug des Smartphones haben Desktop-Rechner aufgehört, in der öffentlichen Wahrnehmung der Inbegriff von Computern zu sein und für das weitere Wachstum digitaler Netze und Apparate zu sorgen. Diese Aufgabe kommt heute einer neuen Klasse von Geräten zu. *Personal computing* verlässt den Schreibtisch im Büro und Heim und wan-

³⁷ Nach dem AT-Formfaktor wurden noch in den 1990er-Jahren PCs mit Pentium-Prozessoren gebaut.

³⁸ Und bis zum heutigen Tag ist die Firmware jedes PCs zum BIOS des IBM PC/AT rückwärtskompatibel.

dert mit Mobiltelefon, Fitness-Tracker und Datenbrille in die Hosentasche, ans Handgelenk und auf den Nasenrücken der Nutzerinnen und Nutzer.

Worin besteht das ‚Gehäuse‘ solcher Computer? (Denn das sind sie, ihrem Aussehen zum Trotz.) Offenkundig sind die ‚Grenzflächen‘ der für den mobilen Gebrauch und die ständige Anbindung ans Internet konzipierten neuen Geräte durch die Infrastruktur der Funknetze und der netzbasierten Anwendungen gegeben. In Hardware wie in Software sind die vernetzten Maschinen dafür ausgelegt, ihren Usern den Zugang zu und den Umgang mit den Diensten von Google, Facebook, Whatsapp, Twitter, Instagram, Snapchat usw. so leicht als möglich zu machen. Im Verein mit Online-Speichern wie iCloud, OneDrive und Dropbox und mit Webapplikationen wie Google Docs, Microsoft Office Online und Adobe Creative Cloud arbeiten sie an der Verwirklichung der Vision einer zeitlich wie räumlich unbegrenzten Verfügbarkeit persönlicher Daten. Und die verschiedenen Techniken, die derzeit unter Industrieschlagworten wie *ubiquitous computing*, *smart environments* und *ambient intelligence* verhandelt werden, wollen darüber hinaus auch die Gebrauchsgegenstände, Räume, Gebäude, Wege und Umgebungen des täglichen Lebens für das Erzeugen, Sammeln, Verteilen und Weiterleiten von Daten dienstbar machen. Den Leitgedanken, der all diesen Unternehmungen als gemeinsame Richtlinie zugrunde liegt, hat Mark Weiser am Ende des letzten Jahrtausends formuliert: „computers embedded in the everyday world“³⁹. Universale diskrete Maschinen sollen aus ihren bisherigen Gehäusen (den Plastik- und Metallverkleidungen von Desktop-Rechnern und Laptops) auswandern und in die Masse der Dinge unserer hochindustrialisierten Gesellschaften, als Implantate gar in unsere Körper eingehen. In letzter Konsequenz würden damit wir selbst, würde die Welt zum ‚Gehäuse‘ eines allumfassenden Computernetzwerks.

Die der hier skizzierten Tendenz immanente geschichtliche Gefahr besteht in ihrer pervertierten Erfüllung, d. h. einer nicht bloß technischen, sondern technokratischen Vollendung: der Möglichkeit, dass ein allumfassendes Computernetzwerk zum ‚Gehäuse‘ der Welt wird – zu einem digitalen Käfig, der uns in einer restlosen Verdattung und Algorithmierung aller Lebensverhältnisse gefangen hält. Im Kleinen ist diese Gefahr der Unfreiheit bereits Wirklichkeit geworden. Bei den digitalen Gadgets von heute handelt es sich nämlich zumeist, sowohl auf der Ebene der Hardware wie der Software, nicht um offene Plattformen (wie noch beim durch Formfaktoren geregelten Komponenten-PC), sondern um weitgehend geschlossene Architekturen.⁴⁰ So beruhte etwa der immense Erfolg von Apples iPod-Reihe nicht zuletzt auf der ‚bedienerfreundlich‘ gestalteten Reduktion der Funktionalität einer eigentlich universell programmierbaren Maschine auf einige wenige mediale Funktionen sowie der

³⁹ Mark Weiser, „The Computer for the 21st Century“, in: *Mobile Computing and Communications Review* 3, 3 (1999), S. 3-11: 5.

⁴⁰ Dass Android, das am weitesten verbreitete Betriebssystem für Smartphones, von Google als Open-Source-Projekt entwickelt wird, ändert an dieser Tatsache kaum etwas.

technischen Verankerung der Hardware in der iTunes-Software und ihrer restriktiven ökonomischen Integration in den zugehörigen iTunes-Store. Denselben Prinzipien der gezielten Verknappung komputationeller Kapazitäten folgen die neuesten iPhones, Kindles und Galaxy Tabs – auch wenn die darauf installierbaren sogenannten Apps den Anschein unbeschränkter Wahlfreiheit erwecken mögen. Die informatische Geschlossenheit dieser *consumer products* – vom Design der GUIs über den Zwang zentral signierter Applikationen bis hin zu den *gated communities* von Facebook und Co. und den häufig nicht zwischen einzelnen Diensten portablen personenbezogenen Daten – zwingt Nutzerinnen und Nutzern das Muster eines rein konsumierenden Umgangs mit und Gebrauchs von Computertechnik auf. Das Problem liegt also nicht allein darin, dass sich die (im herkömmlichen Wortsinne verstandenen) Gehäuse aktueller Geräte, wie eingangs erwähnt, kaum noch öffnen lassen. Mindestens so problematisch ist, dass Eingriffe ins System von den Herstellern überhaupt nur noch in der Form des Bedienens kommerzieller Programme vorgesehen sind. Und was diesen Umstand anbelangt, sind Kittlers Warnungen vor dem ‚Blendwerk‘ von Software und der Beschränkung von Hardware-ressourcen⁴¹ ernst zu nehmen, ja, sind sie drängender denn je. Solche ‚Gehäuse‘ der Computer, deren operativer Zusammenhang von Hard- und Software samt der sozialen, ökonomischen und rechtlichen Ausformung und Einbettung gestaltende Zugriffe der Besitzer auf das System ausschließt, gilt es theoretisch wie praktisch aufzubrechen.

Literatur

- Ceruzzi, Paul E., *A History of Modern Computing*, 2. Aufl., Cambridge, MA, 2003.
- Cringely, Robert X., *Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date*, 2. Aufl., New York, NY, 1996.
- Felsenstein, Lee, „The Tom Swift Terminal or, a Convivial Cybernetic Device“, 1976, online unter: http://www.leefelsenstein.com/wp-content/uploads/2013/01/TST_scan_150.pdf, zuletzt aufgerufen am 11.03.2016.
- Heilmann, Till A., „Die Oberflächlichkeit des Digitalen“, in: Christina Lechtermann/ Stefan Rieger (Hg.), *Das Wissen der Oberfläche. Epistemologie des Horizontalen und Strategien der Benachbarung*, Zürich, Berlin, 2015, S. 253-266.
- Helmers, Carl, „A Nybble on the Apple“, in: *BYTE* 2, 4 (1977), S. 10, online unter: http://archive.org/stream/byte-magazine-1977-04/1977_04_BYTE_02-04_Baudot_Machines, zuletzt aufgerufen am 11.03.2016.
- Isaacson, Walter, *Steve Jobs: The Exclusive Biography*, New York, NY, 2011.

⁴¹ Siehe Kittler (1993), Protected Mode; ders. (1993), Es gibt keine Software.

- Jobs, Steve/Canon, Maggie/Freiberger, Paul, „Apple’s Steven P. Jobs Talks to IW (Interview)“, in: *InfoWorld* 4, 9 (1982), S. 12-15, online unter: <https://books.google.de/books?id=gT4EAAAAMBAJ>, zuletzt aufgerufen am 11.03.2016.
- Kittler, Friedrich (Hg.), *Austreibung des Geistes aus den Geisteswissenschaften. Programme des Poststrukturalismus*, Paderborn, 1980.
- Ders., *Grammophon, Film, Typewriter*, Berlin, 1986.
- Ders., *Optische Medien. Berliner Vorlesung 1999*, Berlin, 2002.
- Ders., „Es gibt keine Software“, in: ders., *Draculas Vermächtnis. Technische Schriften*, Leipzig, 1993, S. 225-242.
- Ders., „Protected Mode“, in: ders., *Draculas Vermächtnis. Technische Schriften*, Leipzig, 1993, S. 208-224.
- Ders., „Hardware, das unbekannte Wesen“, in: Sybille Krämer (Hg.), *Medien, Computer, Realität*, Frankfurt/M., 1998, S. 119-132.
- Levy, Steven, *Hackers: Heroes of the Computer Revolution*, Garden City, NY, 1984.
- Licklider, Joseph C. R., „Man-Computer Symbiosis“, in: *IRE Transactions on Human Factors in Electronics*, 1 (1960), S. 4-11.
- Swaine, Michael/Freiberger, Paul, „Lee Felsenstein: Populist Engineer“, in: *InfoWorld* 5, 45 (1983), S. 105-107, online unter: <https://books.google.de/books?id=0C8EAAAAMBAJ>, zuletzt aufgerufen am 11.03.2016.
- Tholen, Georg Christoph, *Die Zäsur der Medien*, Frankfurt/M., 2002.
- Turing, Alan M., „Rechenmaschinen und Intelligenz“, in: ders., *Intelligence Service. Schriften*, hg. v. Bernhard J. Dotzler und Friedrich Kittler, Berlin, 1987, S. 147-182.
- Ders., „Über berechenbare Zahlen mit einer Anwendung auf das Entscheidungsproblem“, in: ders., *Intelligence Service. Schriften*, hg. v. Bernhard J. Dotzler und Friedrich Kittler, Berlin, 1987, S. 17-60.
- Turner, Fred, *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*, Chicago, IL, 2006.
- Weiser, Mark, „The Computer for the 21st Century“, in: *Mobile Computing and Communications Review* 3, 3 (1999), S. 3-11.
- Wirth, Sabine, „Between Interactivity, Control, and ‚Everydayness‘ – Towards a Theory of User Interfaces“, in: Florian Hadler/Joachim Haupt (Hg.), *Interface Critique*, Berlin, 2016 (im Erscheinen).
- Zopfi, Emil, *Jede Minute kostet 33 Franken*, Zürich, 1977.

ABBILDUNGSNACHWEISE

Till A. Heilmann

Abb. 1 – Wikipedia, online unter: https://en.wikipedia.org/wiki/Apple_II_series#/media/File:Apple_II_advertisement_Dec_1977_page_1.jpg, zuletzt aufgerufen am 02.03.2016, Public Domain.

Abb. 2 – Lawrence Livermore National Laboratory, online unter: <https://www.flickr.com/photos/l1nl/4886601752>, zuletzt aufgerufen am 25.02.2016, Courte-sy of Lawrence Livermore National Laboratory.

Abb. 3 – Dartmouth College, online unter: <https://www.flickr.com/photos/dartmouthflickr/13566201255/in/album-72157643275728555/>, zuletzt aufgerufen am 03.03.2016, Courtesy of Dartmouth College Library.

Abb. 4 – Wikipedia, online unter: https://commons.wikimedia.org/wiki/File:Motherboard_Baby_AT.jpg, zuletzt aufgerufen am 08.03.2016, Creative Commons Attribution-ShareAlike 2.0 Generic.