

Digitale Schriftlichkeit: Programmieren, Prozessieren und Codieren von Schrift

Bartelmus, Martin (Ed.); Nebrig, Alexander (Ed.)

Veröffentlichungsversion / Published Version
Sammelwerk / collection

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:
transcript Verlag

Empfohlene Zitierung / Suggested Citation:

Bartelmus, M., & Nebrig, A. (Hrsg.). (2024). *Digitale Schriftlichkeit: Programmieren, Prozessieren und Codieren von Schrift* (Literatur in der digitalen Gesellschaft, 8). Bielefeld: transcript Verlag. <https://doi.org/10.14361/9783839468135>

Nutzungsbedingungen:

Dieser Text wird unter einer CC BY Lizenz (Namensnennung) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier:
<https://creativecommons.org/licenses/by/4.0/deed.de>

Terms of use:

This document is made available under a CC BY Licence (Attribution). For more information see:
<https://creativecommons.org/licenses/by/4.0>

Martin Bartelmus,
Alexander Nebrig (Hg.)

Digitale Schriftlichkeit

Programmieren, Prozessieren
und Codieren von Schrift

[transcript]

Literatur in der
digitalen Gesellschaft

Martin Bartelmus, Alexander Nebrig (Hg.)
Digitale Schriftlichkeit

Martin Bartelmus ist Postdoc am Institut für Germanistik der Heinrich-Heine-Universität Düsseldorf mit den Forschungsschwerpunkten French Theory, Object-Oriented Ontology, Animal und Plant Studies sowie Materialität, Medialität und Schriftlichkeit.

Alexander Nebrig ist Professor für Neuere Deutsche Literaturwissenschaft an der Heinrich-Heine-Universität Düsseldorf. Zu seinen aktuellen Forschungsgebieten zählen die Geschichte und Theorie der Schrift, die Beziehung der Literatur zur Ethik und die Geschichte des interlingualen Lizenzraumes.

Martin Bartelmus, Alexander Nebrig (Hg.)

Digitale Schriftlichkeit

Programmieren, Prozessieren und Codieren von Schrift

[transcript]

Wir danken für die Unterstützung durch die Heinrich-Heine-Universität Düsseldorf.

Gefördert von der Anton-Betz-Stiftung der Rheinischen Post e.V.



ANTON-BETZ-STIFTUNG
DER RHEINISCHEN POST E.V.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <https://dnb.dnb.de/> abrufbar.



Dieses Werk ist lizenziert unter der Creative Commons Attribution 4.0 Lizenz (BY). Diese Lizenz erlaubt unter Voraussetzung der Namensnennung des Urhebers die Bearbeitung, Vervielfältigung und Verbreitung des Materials in jedem Format oder Medium für beliebige Zwecke, auch kommerziell.

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

Die Bedingungen der Creative-Commons-Lizenz gelten nur für Originalmaterial. Die Wiederverwendung von Material aus anderen Quellen (gekennzeichnet mit Quellenangabe) wie z.B. Schaubilder, Abbildungen, Fotos und Textauszüge erfordert ggf. weitere Nutzungsgenehmigungen durch den jeweiligen Rechteinhaber.

Erschienen 2024 im transcript Verlag, Bielefeld

© **Martin Bartelmus, Alexander Nebrig (Hg.)**

Umschlaggestaltung: Kordula Röckenhaus, Bielefeld; nach einem Entwurf von Martin Bartelmus

Druck: Majuskel Medienproduktion GmbH, Wetzlar

<https://doi.org/10.14361/9783839468135>

Print-ISBN: 978-3-8376-6813-1

PDF-ISBN: 978-3-8394-6813-5

Buchreihen-ISSN: 2750-7610

Buchreihen-eISSN: 2750-7637

Gedruckt auf alterungsbeständigem Papier mit chlorfrei gebleichtem Zellstoff.

Inhalt

Digitale Schriftlichkeit. Eine Einleitung

Martin Bartelmus, Alexander Nebrig 7

Von ›bits‹ zu ›words‹

John von Neumanns *linguistic turn* als Ursprung digitaler Schriftlichkeit

Gabriele Gramelsberger 47

(Elektrographisches) Schreiben als Praxis in Bild, Schrift und Material

Tilo Reifenstein 65

Maschinenschreiben zwischen Energieia und Ergon

Eine sprachphilosophische Untersuchung von Large Language Models

Lucas Falkenhain 83

Schreiben und Lesen als Mensch-Maschine-Kommunikation

Julia Nantke 111

Romantische Maschinen oder: Ein Bericht für ein Literaturhaus

Philipp Schönthaler 125

“What the Heck is A Book”?

Code and Codex in Postdigital Literature

Joachim Harst 149

Transitorische Literatur

Ebenen digitaler Schrift im Blog *Ze zurrealism itzelf*

Lore Knapp und Claus-Michael Schlesinger 171

Vernakulärer Code oder die Geister, die der Algorithmus rief – digitale Schriftlichkeit im Kontext sozialer Medienplattformen

Christian Schulz 189

Humanities of the Digital

Philologische Perspektiven auf Source Codes

als Beitrag einer computerarchäologischen Knowledge Preservation

Stefan Höltgen 207

“I hope you can read this”

Uncovering messages with Critical Code Studies

Mark C. Marino 231

Autor:innen 249

Digitale Schriftlichkeit. Eine Einleitung

Martin Bartelmus, Alexander Nebrig

»L'homme pense et ses actions sont machinales.«¹

In den Literatur- und Sprachwissenschaften meint digitale Schriftlichkeit Praktiken des Schreibens und Lesens in einer digitalen und vernetzten Umgebung, die sich von analogen Schriftpraktiken durch Interaktivität und Multimedialität unterscheiden. Die neuen technischen Möglichkeiten der sozialen Interaktion und der elektronischen Vernetzung von Schrift sowie ihrer Kombination mit anderen Medien sind in dieser Intensität im Analogen nie da gewesen. Der qualitative Unterschied entsteht durch ein drittes Merkmal, das Henning Lobin als Hybridität bezeichnet.² Es bedingt die anderen beiden insofern,

-
- 1 S. Karsakof [d.i. Semen Nikolaievič Korsakov]: *Aperçu d'un procédé nouveau d'investigation au moyen de machines à comparer les idées. Avec deux planches.* St. Petersburg: De l'imprimerie de la III-me section de la chancellerie privée de sa Majesté Impériale 1832, 1. Auf Deutsch lautet der erste Satz der Schrift: »Der Mensch denkt und seine Handlungen sind maschinell.« (Übersetzung von AN, das schwer zugängliche Original wird zitiert nach dem digitalisierten Exemplar der Russischen Staatsbibliothek in Moskau, OR Φ.137 κ.167 ed.36) Vgl. zu Korsakov Wladimir Velminksi/Wolfgang Ernst: *Semën Karsakov. Ideenmaschine. Von der Homöopathie zum Computer.* Berlin: Kadmos 2008, die sich entgegen allen Schreibungen und allein auf der Basis der französischen Transkription des Namens von 1832 für die Schreibung ›Karsakov‹ entscheiden.
 - 2 Genau genommen sind »Automatisierung, Datenintegration und Vernetzung« für Lobin »Triebkräfte[] der Digitalisierung« und »Hybridität, Multimediali-

als Interaktivität und Multimedialität im digitalen Raum ihre beeindruckenden Leistungen aus der Automatisierung der Schrift schöpfen und zu einer Hybridisierung von Mensch und Maschine sowohl beim Lesen als auch beim Schreiben führen. In vorliegender Einleitung konzentrieren wir uns auf dieses Merkmal. Es geht also nicht um digitale Schriftkommunikation von und für Menschen, weder um soziales Lesen noch um multimediale Semantiken, Literatur im digitalen Raum oder Fake News. Vielmehr stehen jene Formen digitaler Schriftlichkeit infrage, die entweder ganz oder teilweise auf maschineller Seite zu verorten sind: Programme, Codes und Algorithmen. Menschen nutzen die digitale Schreib- und Lesetechnik nicht nur, sondern gestalten sie mit und überantworten Schreibprozesse den Maschinen. Digitale Schriftlichkeit menschlicher Akteure:innen basiert darauf, dass Maschinen lesen und schreiben, dass maschinelle Schriftlichkeit unseren Raum des Les- und Schreibbaren strukturiert. Es gibt eine Schriftlichkeit der digitalen Schriftlichkeit, eine Potenzierung, die analogen Schriftkulturen unbekannt ist.

Walter Ong deutet die Erfindung der Schrift als Technologisierung der Sprache.³ Die Gleichsetzung von Mündlichkeit mit Natürlichkeit und Schriftlichkeit mit Künstlichkeit erfolgt auf der Annahme, dass orale Sprache nicht als Kunst im Sinne einer Technik erlernt werden muss, sondern sich wie von selbst in jedem Individuum entwickelt. Schrift dagegen impliziert eine technische Theorie der Sprache. Für die digitale Schriftlichkeit muss Ongs These weiter differenziert werden. Denn die analoge Technologisierung der Rede als Schrift hat anders als ihre digitale Fortsetzung noch keinen doppelten Boden, ist eine Technik erster Ordnung. Anders gesagt: Digitale Schrift funktioniert nur deshalb, weil sie eine weitere Schriftebene schafft, welche die sichtbare

tät und Sozialität« von jenen bewirkte Tendenzen (Henning Lobin: *Engelbarts Traum. Wie der Computer uns Lesen und Schreiben abnimmt*. Frankfurt a.M./New York: Campus Verlag 2014, 96).

3 Walter J. Ong: *Orality and Literacy. The Technologizing of the Word*. London: Methuen & Co 1982.

Schrift im Besonderen und die digitalen Medien im Allgemeinen nochmals technisiert. Es kommt neben der Technologisierung der Sprache zu einer Technologisierung der Schrift.

Diese Schrifttechnik zweiter Ordnung ist menschengemacht, verändert die Menschen und vollzieht Operationen, an denen sie nicht beteiligt sind. Somit kristallisieren sich drei Aspekte digitaler Schriftlichkeit heraus: Erstens erfinden Menschen Schriften, um mit Maschinen zu kommunizieren und sie in ihren Dienst zu nehmen. Programme werden geschrieben und gelesen, um Computer zu benutzen. Zweitens gibt es eine Form digitaler Schriftlichkeit, die sich ganz ohne menschliches Zutun vollzieht, da Maschinen auch mit Maschinen kommunizieren und Prozesse ausführen, damit wir lesen und schreiben können. Drittens geht es um die alltägliche Erfahrung, dass der digitale Code zwar menschengemacht ist, aber seine eigenen Affordanzen besitzt. Maschinen adressieren ihre menschlichen Nutzer:innen, Algorithmen ›programmieren‹ unser Verhalten. Unter diesen drei Aspekten sollen im Folgenden die drei Grundoperationen digitaler Schriftlichkeit vorgestellt werden. Programmieren betrifft die Mensch-Maschine-Kommunikation, Prozessieren maschinelle Vorgänge des Lesens bzw. Auslesens und Codieren die dem digitalen Code inhärente Eigenlogik, wie sie im Algorithmus zur Sprache kommt.

1. Programmieren: Mit Maschinen sprechen

Der Turing-Test, ob diese Einleitung maschinell verfasst ist, kann mittlerweile selbst wieder von einer Maschine durchgeführt werden. Die Leistungsfähigkeit, die Textgeneratoren in den letzten Jahren erreicht haben, zeigt sich darin, dass auch Künstliche Intelligenz (KI) eine Schwierigkeit damit hätte, zu beweisen, ob ein Text von einem Chatbot oder einem Menschen geschrieben worden ist. Es ist nicht mehr

erkennbar, ob Übersetzer:innen oder Autor:innen aus Fleisch und Blut einen Text verantworten oder eine KI.⁴

Als erste Annäherung an das Programmieren, das eine zentrale Praxis auf dem Gebiet der digitalen Schriftlichkeit darstellt, erweist sich Alan Turing noch in einer weiteren Hinsicht als Impulsgeber. Mit seinem Konzept der *paper machine*⁵ begreift er nicht die Maschine als Menschen, sondern den Menschen als Maschine. Es sei möglich, so Turing 1948,⁶ eine Rechenmaschine nachzuahmen, indem man Handlungsanweisungen aufschreibt und sie von einem Menschen ausführen lässt. Eine solche Verbindung von Menschen und aufgeschriebenen Anweisungen, die man auch kurz Programm nennen kann, bezeichnet Turing als Papiermaschine:

»Es ist möglich, den Effekt einer Rechenmaschine zu erreichen, indem man eine Liste von Handlungsanweisungen niederschreibt und einen Menschen bittet, sie auszuführen. Eine derartige Kombination eines Menschen mit geschriebenen Instruktionen wird ›Papiermaschine‹ genannt. Ein Mensch, ausgestattet mit Papier, Bleistift und Radiergummi sowie strikter Disziplin unterworfen, ist in der Tat eine Universalmaschine.«⁷

4 Das hat natürlich auch Konsequenzen für das menschliche Schreiben. Vgl. Crystal Chokshi: In Other Words: Smart Compose and the Consequences of Writing in the Age of AI. In: *Culture Maschine* 20 (2021): <https://culutremachine.net/vol-20-machine-intelligences/> (zuletzt abgerufen am 21.12.2023); *Die Herausgeber haben sich nach besten Wissen und Gewissen bemüht, sicherzustellen, dass die URLs für externe Websites, auf die in diesem Buch verwiesen werden, zum Zeitpunkt der Drucklegung korrekt und aktiv sind. Die Herausgeber übernehmen jedoch keine Verantwortung für die Websites und können keine Garantie dafür geben, dass eine Website aktiv bleibt oder dass der Inhalt angemessen ist oder bleiben wird.*

5 Vgl. ferner die Variationen der Turing-Maschine: Oswald Wiener/Manuel Bonik/Robert Hödicke: *Eine elementare Einführung in die Theorie der Turing-Maschinen*. Wien/New York: Springer 1998.

6 Alan Turing: Intelligente Maschinen. In: ders.: *Intelligence Service. Schriften*. Hg. von Bernhard J. Dotzler und Friedrich Kittler: Berlin: Brinkmann & Bose 1987, 81–113.

7 Turing: Maschinen, 91.

Die Papiermaschine zeichnet sich aus durch ihren Universalismus, endloses Papier und unendlich viel Tinte.⁸ Man könnte versucht sein, mit der Papiermaschine eine:n Schauspieler:in zu assoziieren, der/die das Stück eines/einer Dramatikers:in aufführt. Ist eine Tragödie ein Programm, eine niedergeschriebene Instruktion? Die Frage lässt sich ebenso auf eine musikalische Komposition und ihre Aufführung beziehen. Doch es besteht ein Unterschied. Ein niedergeschriebenes Drama wird nicht ausgeführt, sondern aufgeführt. Die Performanz ist dem Programm, das allein syntaktische Regeln befolgt, unbekannt. Till Heilmann weist darauf hin, dass dem Programm die »pragmatische und semiotische Dimension« der Schrift abgehe:

»Erst wenn eine Schrift auf ihre bloßen syntaktischen Formen reduziert ist und diese Formen durch physikalische Zustände, beispielsweise als Spannungspotentiale in Transistoren, »anschreibbar« sind, können sie in einem Computer die regelgeleiteten Kaskaden von Schaltvorgängen auslösen, an deren Ende der durch eben diese Schaltvorgänge veränderte Zustand des Computers als Ergebnis der Programmierung »ablesbar« ist.«⁹

Ohne Code kein Klang, keine Sprache, keine Schrift und keine Bilder, die auf Bildschirmen erscheinen oder von Computern ausgegeben werden können. Code ist eine Art *Lingua franca* geworden, »not only of computers but of all physical reality«, wie N. Katherine Hayles konstatiert.¹⁰ Die Turing-Maschine steht für sie am Anfang eines »computational re-

8 Bernhard J. Dotzler: *Papiermaschine. Versuch über COMMUNICATION & CONTROL in Literatur und Technik*. Berlin: Akademie Verlag 1996, 7.

9 Till A. Heilmann: *Textverarbeitung. Eine Mediengeschichte des Computers als Schreibmaschine*. Bielefeld: transcript 2012, 65.

10 N. Katherine Hayles: *My Mother Was a Computer. Digital Subjects and Literary Texts*. Chicago/London: The University of Chicago Press 2005, 15.

gime«,¹¹ in dem und durch das Code ontologisch wird und das gleichzeitig »as metaphor and means« funktioniert.¹²

»Computation is a scriptorial process, and the computer is a scripting machine«,¹³ so Louisa Shen, und auch Florian Cramer versteht Computerprogramme als Sprache und Schrift erzeugende Maschinen auf der Basis von Codes: »The digital computer is a symbolic machine that computes syntactical language and processes alphanumerical symbols; it treats all data – including images and sounds – as textual, that is, as chunks of coded symbols.«¹⁴ Steven Connor bestimmt ferner Schreiben als eine Art der Maschinerie und fragt: »If all writing is a kind of machinery, why might it be plausible to see every machine as a kind of writing?«¹⁵ Schreiben basiert auf einer Differenz von natürlicher und maschineller Sprache,¹⁶ wobei letztere nicht ohne erstere denkbar und zudem ihrerseits aufgeteilt ist in »languages in which algorithms are expressed and software is implemented.«¹⁷ Für die Mensch-Maschine-Kommunikation sind ferner sogenannte *Assembler*, also Übersetzungsprogramme notwendig, wobei »ein Programm in Assemblersprache sehr einem Programm in Maschinensprache [ähnel], das für Menschen lesbar gemacht worden ist.«¹⁸ *Compiler* wiederum sind Programme, die mehr Übersetzungsarbeit leisten. *Interpreter* dagegen übersetzen nicht mehr in Maschinensprache, sondern »lesen eine Zeile und führen

11 Hayles: *My Mother*, 18. Hayles geht dabei von Stephann Wolframs Buch *A New Kind of Science* aus.

12 Hayles: *My Mother*, 20f.

13 Louisa Shen: *GUilding the Overwrite*. In: James Gabrillo/Nathaniel Zetter (Hg.): *Articulating Media. Genealogy, Interface, Situation*. Open Humanities Press 2023, 78–95, hier: 78.

14 Florian Cramer: *Language*. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. Cambridge MA/London: MIT Press, 168–173, hier: 171.

15 Steven Connor: *How to Do things with Writing Machines*. In: Sean Pryor/David Trotter (Hg.): *Writing, Medium, Machine. Modern Technographies*. O.O.: Open Humanities Press 2016, 18–34, hier: 18.

16 Vgl. für eine kurze Skizze der Maschinensprache: Douglas R. Hofstadter: *Gödel, Escher, Bach ein Endloses Geflochtenes Band*. München: DTV 92003, 310f.

17 Cramer: *Language*, 168.

18 Hofstadter: *Gödel, Escher, Bach*, 312.

sie sofort aus.«¹⁹ Das hat zur Folge, dass Schriftlichkeit der digitalen Schriftlichkeit, also die Technisierung der Schrift, die Zeichen der Welt weiter umcodiert: »Verhalf das Alphabet dazu, den Körper der Welt (Soma) in ein Zeichen (Sema) zu verwandeln, geht der digitale Code weiter und löst nun auch die Welt der Zeichen (Sema) in Digits (bits) auf.«²⁰

Was den bzw. die Schauspieler:in – um weiter im Kontext von Körper und Code zu bleiben – vom Computer unterscheidet, und damit auch ganz grundsätzlich alle Rechenmaschinen vor Turings Papiermaschine von denen von heute, ist eine »Auftrennung in Funktionsgruppen«. ²¹ Für Bernhard Dotzler beginnt der Computer dort, »wo Funktionsgruppen sich aufspalten.«²² Diese Auftrennung, die nicht vom Körper der Schauspieler:in geleistet wird, bezeichnet die Funktionen: Rechnen, Speichern, Übertragen – Computing, Control, Communication.²³

In diesem Kontext kann der Untertitel des vorliegenden Bandes – *Programmieren, Prozessieren und Codieren von Schrift* – auch anders formuliert werden: Vorschreiben, Fortschreiben und Einschreiben. Diese Trias meint nicht einfach nur »Textverarbeitung«, also das Schreiben normalsprachlicher Texte mithilfe des Computers.²⁴ Es markiert das Weichwerden der Wörter als ihre Codier- und Decodierbarkeit. Damit geht die Unterscheidung von »elektrisch« und »elektronisch« einher. Letzteres meint, dass elektrisch codiert und decodiert wird.²⁵ Diskrete bzw. digitale Schriftlichkeit ist nicht auf mechanisches, sondern auf maschinenartiges Schreiben bezogen, auf die Schrift einer idealen Maschine.²⁶

19 Hofstadter: *Gödel, Escher, Bach*, 314.

20 Martin Burkhardt/Dirk Höfer: *Alles und Nichts. Ein Pandämonium digitaler Weltvernichtung*. Matthes & Seitz: Berlin 2015, 49.

21 Vgl. Dotzler: *Papiermaschine*, 32.

22 Dotzler: *Papiermaschine*, 30.

23 Vgl. Dotzler: *Papiermaschine*, 41.

24 Textverarbeitung bzw. »word-processing« war ab 1967 als Begriff in Umlauf und wurde durch IBMs 1961 vorgestellten »electric typewriter« populär gemacht. Vgl. Connor: *Writing Machines*, 28.

25 Connor: *Writing Machines*, 29.

26 Vgl. Connor: *Writing Machines*, 31f.

Digitale Schriftlichkeit forciert die skripturalen Prozesse in und an der Maschine und weniger Be-Schreibungen wie Franz Reuleaux' *Kinematische Zeichensprache* oder Charles Babbages *Mechanical Notation*, zwei Versuche, »Maschinen als das mechanische Getriebe, das sie sind, zu notieren«. ²⁷ Semën N. Korsakov, als Gegenentwurf zu Babbage, entwarf 1832 »ein Lochkartenverfahren [...], das Datenverarbeitung zum Zweck von *intelligence* einsetzt«, ²⁸ und konzipierte damit eine Ideenmaschine. Ziel ist eine »begriffsvergleichende Maschine«, die »ganz im Sinne Alan Turings eine scheinbar geistige intellektuelle Tätigkeit vollständig durch ein mechanisches Äquivalent« ersetzt. ²⁹ Auch für diese Maschine bedarf es einer Programmierung, folglich einer Anschrift mithilfe von »Datenbanken und Tabellen«. ³⁰

Die digitale Schriftlichkeit versteht sich demnach als eine Form der Schriftlichkeit, die sich jenseits der technischen Vorläufer von digitalen Computern, den *Machinae Arithmeticae*, verortet und sich dezidiert als prozessuale Materialität der Schreibweise einer neuen Konfiguration zwischen ›Control‹ und ›Communication‹ darstellen lässt. ³¹ Damit schließt die digitale Schriftlichkeit an die Schreibstuben und ihre Begriffe ›Kontor/Comptoir‹ an, ³² wird als Funktion der Buchführung sichtbar in dem Sinne, dass Buchführung eben »Grundelement schriftbasierter Gesellschaften« ist. ³³ Von hier aus ist es nicht mehr weit bis zum Kernbegriff des Computerzeitalters »computus« (als Programmierkunst) und dann zum »Zauberwort« Algorithmus. ³⁴

27 Dotzler: *Papiermaschine*, 48.

28 Wolfgang Ernst: Ein medienarchäologisches Schicksal. In: Wladimir Velminski/Wolfgang Ernst: *Semën Karsakov. Ideenmaschine. Von der Homöopathie zum Computer*. Berlin: Kadmos 2008, 7–48, hier: 20.

29 Ernst: *Schicksal*, 27.

30 Ernst: *Schicksal*, 27.

31 Angeregt von N. Catherine Hayles: *Print is Flat, Code is Deep: The Importance of Media-Specific Analysis*. In: *Poetics Today* 25/1 (2004), 67–90.

32 Dotzler: *Papiermaschine*, 66.

33 Dotzler: *Papiermaschine*, 66.

34 Vgl. Dotzler: *Papiermaschine*, 69.

Schriftlichkeit gehört existenziell zum Diskurs um Datenverarbeitung, in dessen Zentrum das Wort Information steht. Information ist nicht nur Nebenprodukt von Prozessen, sondern gleichsam auch abstrakt und messbar.³⁵ Claude Shannon wird gemeinsam mit Warren Weaver in *The Mathematical Theory of Communication* die Disziplin der Informationstheorie begründen,³⁶ indem er das Problem des Rauschens in elektronischer Kommunikation als Ausgangspunkt nimmt, um Information an Medialität zu knüpfen.³⁷ Gregory Bateson definiert Information als die Differenz, die Differenz macht.³⁸

Programmieren, Prozessieren und Codieren gehören zum Spektrum um ›Control‹ und ›Communication‹ und schließen den Befehl,³⁹ also ›Command‹ mit ein. Aufgrund der Befehlsstruktur wird die Botschaft ohne jegliches Mitdenken vom maschinellen Empfänger verstanden, als wäre er ein ›Vollidiot‹: »Writing in a computer programming language is phrasing instructions for an utter idiot.«⁴⁰ Der Programmierspezialist Eben Moglen, der in den 1970er Jahren für IBM arbeitete, wertet das Programmieren der Kommandozeile dagegen mit der Bemerkung auf: »What we were doing with computers was making languages that were better than natural languages for procedural thought.«⁴¹

Dadurch ist – im wahrsten Sinne des Wortes – ein für die Schriftlichkeit bestimmender Diskurs angesprochen. Denn Sprache und Sprechen

35 Vgl. Ted Byfield: Information. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. Cambridge MA/London: MIT Press, 125–131, hier: 126.

36 Vgl. dazu: Claude E. Shannon: *Ein/Aus. Ausgewählte Schriften zur Kommunikations- und Nachrichtentheorie*. Hg. von Friedrich Kittler/Peter Berz/David Hauptmann/Axel Roch. Berlin: Brinkmann & Bose 2000.

37 Byfield: Information, 128.

38 Gregory Bateson: *Ökologie des Geistes. Anthropologische, psychologische, biologische und epistemologische Perspektiven*. Frankfurt a.M.: Suhrkamp 1985, 488. Vgl. Jacques Derrida führt seinen Begriff der *différance* als »irreduzible Abwesenheit der Intention« ein, um »die allgemeine graphematische Struktur jeder ›Kommunikation‹ zu behauptet.« Vgl. Jacques Derrida: *Limited Inc*. Wien: Passagen Verlag² 2023, 41.

39 Vgl. Hofstadter: *Gödel, Escher, Bach*, 310.

40 Cramer: Language, 171.

41 Shen: GUIding, 88. (Moglen and Worthington 2000).

vom Befehl her zu denken, wie es in der Geistesgeschichte kontinuierlich versucht wird, markiert die Opposition von Stimme und Schrift, Schriftlichkeit und Mündlichkeit, in deren Zentrum der Phonologozentrismus erscheint, und bildet gleichzeitig den Horizont, vor dem sich eine digitale Version dieser Opposition abzeichnet. Der Futurist Velimir Chlebnikov imaginiert 1916: »Wir können jede Handlung, jede Form mit einer Zahl benennen, und durch das Erscheinenlassen einer Zahl auf dem Glas eines Lämpchens können wir sprechen. [...] Besonders geeignet ist die Zahlensprache für Radiotelegramme. Zahlenreden.«⁴² Dank der Technologie haben wir es heute mit einem arithmetischen Phonologozentrismus zu tun.

Die »Kommandostruktur« der digitalen Kommunikation mit Computern basiert nicht nur auf Kommandozeilen wie bei MS-DOS, sondern geht auf die Lochkarten der Webstühle zurück. Dadurch sind der digitalen Schriftlichkeit eine Form der pastoralen Weisung, einer *Talking Cure*, aber auch ein Sprechakt eingeschrieben, die sich nun nicht mehr unmittelbar an Menschen, sondern an Maschinen richten. Diese überführen die Menschen aus der Disziplargesellschaft in eine Kontrollgesellschaft.⁴³ Programmieren ist voll von einer Sprache der Rede und nicht einer der Schrift: »call, query, confirm, dialogue.«⁴⁴ Gleichzeitig sind diese mündlichen Sprechakte als Diktate rückübersetzt in das schriftliche Kommando und machen eine Redesimulation sichtbar: »Beschreiben und Schreiben wird eins: Programmieren.«⁴⁵

Die Programmierkunst, auf die sich die digitale Schriftlichkeit bezieht, ist auch historisch in gewisser Weise neu und unterscheidet sich von arithmetischen Maschinen und anderen Befehlsketten. Sie ist gekennzeichnet durch die Programmierbarkeit, wie sie als erstes von Ada Lovelace in Bezug auf die Analytical Engine (1833/34) von Babbage

42 Velimir Chlebnikov: Brief an zwei Japaner, In: ders.: *Werke*. Teil 2. Berlin: Suhrkamp 2022, 245–248, hier: 247f.

43 Gilles Deleuze: Postskriptum über die Kontrollgesellschaften. In: ders.: *Unterhandlungen 1972–1990*. Frankfurt a.M.: Suhrkamp ⁵2014, 254–262.

44 Shen: GÜlding, 92.

45 Burkhardt/Höfer: *Alles und Nichts*, 19.

beschrieben wurde.⁴⁶ Programmierbarkeit bedeutet hier die »discretization (or disciplining) of hardware.«⁴⁷ Daraus leitet sich eine Form des Genießens ab, die sich im Schreiben von Programmen offenbart und sich als Macht(-gefühl) oder Souveränitätssimulation der Programmierer:innen äußert.⁴⁸

Donald E. Knuths 1984 vorgestelltes *literate programming* kehrt die Vorzeichen des Genießens beim Programmieren um. Das Adjektiv »literate« bedeutet in diesem Kontext, dass Literatur anders dokumentiert (wird) als Code. Knuth, der eine bessere Dokumentation von Programmen fordert,⁴⁹ wirbt für einen *human turn* in der Informatik: »let us concentrate rather on explaining to *human beings* what we want a computer to do.«⁵⁰ Seit Lovelace und Turing ist also viel passiert: Der Computer muss nicht mehr angeschrieben werden, sondern (erneut) der Mensch. Menschen müssen darüber informiert werden, wie das ominöse ›Wir‹ den Computer dazu bringt, das zu tun, was das Wir will. Es geht nicht nur um das Kommentieren von Code, sondern auch um das Einbinden der Erklärung und des Codes in eine spezifische literarische, sogar philosophisch-politische Form: den Essay.⁵¹ Im Zentrum des Essays steht – jedenfalls für Knuth – die Konstitution eines künstlerischen Subjekts: »I suddenly have a collection of programs that seem quite beautiful to my own eyes, and I have a compelling urge

46 Dotzler: *Papiermaschine*, 536

47 Wendy Hui Kyong Chun: Programmability. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. Cambridge MA/London: MIT Press, 224–228, hier: 225.

48 Chun: Programmability, 227.

49 Donald E. Knuth: Literate Programming. In: *The Computer Journal* 27.2 (1984), 97–111, hier: 97.

50 Knuth: Literate Programming, 97.

51 Auch Mark Marino geht in seiner Konzeption der Critical Code Studies auf Don Knuth ein und versteht das Programme-Schreiben als essayistische Praktik. Vgl. Mark C. Marino: *Critical Code Studies*. Cambridge MA: MIT Press 2020, 41. Auch Max Bense interessiert sich für die Form des Essays. Vgl. Max Bense: *Über den Essay und seine Prosa*. In: *Merkur* 3 (1947): <https://www.merkur-zeitschrift.de/max-bense-ueber-den-essay-und-seine-prosa-75-jahre-merkur/> (zuletzt abgerufen am 15.11.2023).

to publish all of them so that everybody can admire these works of art.«⁵² Der Glaube an die Literarizität der Programmiersprache, also ihre Fähigkeit, Literatur zu werden, wird mithilfe einer ontologischen Konzeption als »idea« begründet.⁵³ Der/die Programmierer:in hat die Metamorphose zum/r Künstler:in beinahe überstanden. Beinahe, denn Knuths Programme wuchern, werden immer mehr, sodass der Speicher nicht mehr ausreicht: »my office will be encrusted with webs of my own making.«⁵⁴ Die Konsequenz ist, in die Welt auszufern, in Zeitschriften zu diffundieren und damit medien-archäologisch eine neue Gedächtnis- und Erinnerungsschicht zu produzieren.

Mit dem neuen Schreiben geht ein spezifischer Genuss (»joy«⁵⁵) einher, der vorher – beim traditionellen Coding/Programmieren – nicht in dieser Form bestanden hat. Der Genuss führt dazu, dass Knuth in der Lage ist, »to write programs as they should be written.«⁵⁶ Wie sollen also Programme geschrieben werden? »My programs are not only explained better than ever before; they also are better programs, because the new methodology encourages me to do a better job.«⁵⁷ Das »better« bleibt unbestimmt, und gleichzeitig impliziert das »literate« sein Gegenteil als moralisches Totschlagargument: »surely nobody wants to admit writing an *illiterate* program.«⁵⁸ Auch Programmierer:innen stehen also in einem Aufschreibesystem, das ihre Lese- und Schreibfähigkeit seinerseits codiert, normiert und kontrolliert.

Begriffsgenealogisch gehen dem Programmieren die Begriffe »Coding« oder »planning the computation« voraus.⁵⁹ Das Codieren meint, dass Information, die codiert werden soll, digital zur Verfügung steht, d.h., sie muss »diskretisiert« und »quantisiert« werden.⁶⁰ Oder anders

52 Knuth: *Literate Programming*, 109.

53 Knuth: *Literate Programming*, 111.

54 Knuth: *Literate Programming*, 109.

55 Knuth: *Literate Programming*, 97.

56 Knuth: *Literate Programming*, 97.

57 Knuth: *Literate Programming*, 97.

58 Knuth: *Literate Programming*, 97.

59 Heilmann: *Textverarbeitung*, 62.

60 Heilmann: *Textverarbeitung*, 196.

formuliert: Wir zählen abzählbare Zeichen,⁶¹ damit diese wiederum gespeichert, übertragen und verarbeitet, d.h. prozessiert werden können. Dabei ist entscheidend, dass nicht nur die gängigen Buchstaben digitalisiert werden, sondern auch das Leerzeichen, Satzzeichen und andere Befehle.

Frieder Nake beschreibt Zeichen im Anschluss an Charles Sanders Peirce als algorithmische Zeichen.⁶² Das hat zur Folge, dass das Zeichen für Mensch und Maschine prozessier- bzw. lesbar wird.⁶³ Wie Hayles bemerkt, haben wir es mit »flackernden Signifikanten« zu tun, einer wie Till A. Heilmann erklärt, ständig changierenden Kette durch maschinelle En- und Decodierprozesse verknüpfter Markierungen.⁶⁴ Diese flackernden Zeichen sind nicht (nur) metaphorisch zu verstehen, sondern auch materiell. Für Turing »flackerten« Zeichen und Text noch nicht.⁶⁵

Mit dem Flackern der Zeichen geht eine neue ontologische Dimension einher, die Auswirkungen auf das Schreiben als Programmieren hat. Die Frage, wer was schreibt, wenn programmiert wird, ist insofern relevant, als sich mit der Programmierbarkeit »die Geschichte des Worts in die der Idee des Algorithmus« verwandelte.⁶⁶ Der/die Programmierer:in erscheint als Autor:in eines Programms, das eine gewisse Abgeschlossenheit und damit Werkcharakter erhält. Der/die Coder:in dagegen liefert variable Codesegmente, die sowohl Programmierer:innen als auch Computer nutzen können. Denn der Code kann auch einfach hinterlegt werden, sodass der Computer je nach Programm auf diese Codesegmente zurückgreift und damit eigenständig »schreibt«. Hardware und Soft-

61 Heilmann: *Textverarbeitung*, 199.

62 Heilmann: *Textverarbeitung*, 230 und Frieder Nake: Das algorithmische Zeichen und die Maschine. In: Hansjürgen Paul/Erich Latniak (Hg.): *Perspektiven der Gestaltung von Arbeit und Technik. Festschrift für Peter Brödner*. München: Rainer Hampp 2004, 203–221.

63 Heilmann: *Textverarbeitung*, 231.

64 Heilmann: *Textverarbeitung*, 217f.

65 James Purdon: Teletype. In: Sean Pryor/David Trotter (Hg.): *Writing, Medium, Machine. Modern Technographies*. O.O.: Open Humanities Press 2016, 120–136, hier: 133.

66 Dotzler: *Papiermaschine*, 71.

ware lassen sich dadurch unterscheiden, aber auch ineinander übersetzen.

Computer können nicht nur mit digitalem Code programmiert werden, sondern auch mit auf Papier gelochten Programmen. Erst mit Textterminals ab den 1960er Jahren und damit mit sogenannten Editoren lässt sich Code für Programme am und mit bzw. für den Computer schreiben. Das bedeutet, dass sich Codes in ihrer Komplexität unterscheiden. Sogenannte höhere Programmiersprachen schaffen immer mehr Distanz zwischen Mensch und Maschine, auch wenn sie die Kommunikation deutlich steigern. Ihre Idealisierung erfährt die absolute Kommunikation dann in der Kybernetik,⁶⁷ die, wie Norbert Wieners Publikation von 1948 im Titel formuliert, Kontrolle und Kommunikation aller nichtmenschlicher Aktanten regelt: *Cybernetics or Control and Communication in the Animal and the Machine*.

Dafür braucht es einen Übersetzer, sogenannte Compiler, die Quellcode in den Objektcode übertragen. Wir haben es meistens mit zwei Textsorten zu tun: dem Programm als Text, also dem Quelltext, sowie dem Maschinencode.⁶⁸ Entweder man sagt wie Jörg Pflüger, dass im Anfang »der Text« war,⁶⁹ oder man sieht wie Jean Baudrillard mit dem Computer die Ära des Codes beginnen: »[T]he era of the signified and of the function is over: it is the era of the signifier and of the code that begins.«⁷⁰ Dieser Befund, den Baudrillard 1972 in einem ökologischen Kontext vorstellt, betont gleichzeitig die Verschiebung von einer analogen zu einer digitalen Schriftlichkeit. Anstelle von Fortschritt sind

67 Vgl. Peter Janich: *Was ist Information?* Frankfurt a. M.: Suhrkamp 2006, 48–57 und zur Geschichte der Kybernetik: Thomas Rid: *Maschinendämmerung. Eine kurze Geschichte der Kybernetik*. Berlin: Propyläen 2016.

68 Heilmann: *Textverarbeitung*, 94.

69 Jörg Pflüger: Writing, Building, Growing. Leitvorstellungen der Programmiergeschichte. In: Hans Dieter Heilige (Hg.): *Geschichten der Informatik. Visionen, Paradigmen, Leitmotive*. Berlin: Springer 2004, 275–320, hier: 281.

70 Jean Baudrillard: Design and Environment. Or, The Inflationary Curve of Political Economy. In: *The Universitas Project. Solutions for a post-technological Society*. New York: Museum of Modern Art 2006, 50–66, hier: 61.

Produkte wie ChatGPT nur Aktualisierungen von theoretischen Vorschriften, die vor mehr als 50 Jahren verfasst wurden. Ähnliches gilt für den *Data Driven Turn* der Digital Humanities. Henning Schmidgen und Bernhard Dotzler weisen in ihrem Buch *Foucault digital* auf den Umstand hin, dass die Geschichtswissenschaft schon 1969 mit Daten arbeitete. Michel Pêcheuxs *Analyse automatique du discours* dachte nicht nur an die Datenverarbeitung, sondern auch an eine Maschine, »die in einem emphatischen Wortsinn zu *lesen* und *deshalb* die Menschen, die »ein Erlernen des Lesens« hinter sich haben, zu ersetzen imstande ist.«⁷¹ Baudrillard scheint auf einen solchen Diskurs hinzuweisen und damit diesen als Lektüresimulation aufzudecken, der sich hinter den Verheißungen der Kybernetik verbirgt.

2. Prozessieren: Eine andere Form des Lesens

Im Zentrum digitaler Schriftlichkeit steht immer wieder die Mensch-Maschine-Kommunikation, aber auch die Kommunikation zwischen Maschinen. Programme, Codes bzw. Software dienen der Vermittlung und Übersetzung. Für diese Kommunikation bedarf es einer eigenständigen Materialität von Interfaces, die aus Keyboards,⁷² Bildschirmen, spezifischen Kathodenstrahlröhren, Lichtstiften,⁷³ der Entertaste,⁷⁴ der Maus, aber auch aus einfachen Lochkarten aus Papier bestehen kann.⁷⁵ Prozessieren bzw. Maschinenlesbarkeit ist stets das Ziel. Der Fokus liegt nicht allein auf der oder dem *User:in*, sondern auf der Maschine,

71 Bernhard J. Dotzler/Henning Schmidgen: *Foucault, digital*. Lüneburg: Meson Press 2022, 71.

72 Shen: GÜlding, 79: »The keyboard has been a mainstay of programming and processing, as has paper (albeit with varying degrees of importance as memory and display components have evolved).«

73 Shen GÜlding, 80.

74 Connor: *Writing Maschine*, 30.

75 Vgl. zu der Materialität digitaler Speichermedien: Wolfgang Ernst: *Das Gesetz des Gedächtnisses. Medien und Archive am Ende (des 20. Jahrhunderts)*. Berlin: Kadmos 2007, 205–260.

die benutzt wird.⁷⁶ Maschinen, vor allem vernetzte, automatisieren Kommunikationsprozesse nicht einfach, sie vervielfachen diese auch, sodass Sadie Plant konstatiert: »paralleles Prozessieren automatischer Kommunikation, verschalteter Leitungen, wiederholter Operationen; Muster und Netzwerke, die sich wie Unkraut vermehren.«⁷⁷ Alles in allem erscheint dabei der Begriff der digitalen Kommunikation dürftig – eine Chance für die digitale Schriftlichkeit.

Programmieren, Prozessieren und Codieren, also die maschinenartigen Schreib- und Leseprozesse, die nicht nur auf Menschen, sondern auch und besonders auf Maschinen bezogen sind, gehören einer Ordnung der Grammatologie an, die Catherine Malabou als allgegenwärtiges ›motor scheme‹ bezeichnet.⁷⁸ Darin wäre digitaler Computercode eine Art ›worldview‹, wie Hayles erklärt, die zugleich Funktion und Bedeutung von Schrift und Sprache beeinflusst und *vice versa*.⁷⁹ Grammatologie und ›Computation‹, so Hayles, stehen sich aber gegenüber.⁸⁰

Von Ada Lovelace bis Knuth spielen die klassisch literaturwissenschaftlichen Kategorien von Autor:innenschaft,⁸¹ Werk und Schrift zentrale Rollen, die sich im Zeichen des *motor schemes* ›Grammatologie‹ mithilfe von *Rhetorical Code Studies*⁸² markieren und zwischen *Critical* und *Source Code Studies* in Hinblick auf die Frage nach einer prozessontologischen digitalen Schriftlichkeit neu verorten lassen können und auf die Begriffe Code, Programme, Software und Algorithmus ausgreifen. Dabei lassen sich Programmieren, Prozessieren und Codieren mit den

76 Connor: Writing Maschine, 31.

77 Sadie Plant: *nullen + einsen. Digitale Frauen und die Kultur der neuen Technologien*. Berlin Verlag 1998, 122.

78 Wir verweisen hier auf unsere Einleitung zum Band *Schriftlichkeit. Agentialität, Aktivität und Aktanten von Schrift*. Bielefeld: transcript 2022.

79 Hayles: *My Mother*, 16.

80 Hayles: *My Mother*, 17.

81 Vgl. Kathleen Fitzpatrick: The Digital Future of Authorship. Rethinking Originality. In: *Culture Machine* 12 (2011): <https://culturemachine.net/the-digital-humanities-beyond-computing/> (zuletzt abgerufen am 21.12.2023).

82 Vgl. Kevin Brock: *Rhetorical Code Studies. Discovering Arguments in and around Code*. Ann Arbor: University of Michigan Press 2019.

textuellen Ebenen von Bedeutung, Lesbarkeit und Zeichenhaftigkeit parallelisieren.

Prozessieren verweist dabei auf eine Lesbarkeit und ein Lesen, mit- hin auf eine vermeintliche Widerstandslosigkeit des programmierten Codes. Prozessieren hängt zudem spezifisch vom »compiling« ab, also der Interpretation des Codes.⁸³ Dazu bedarf es auch neuer Praktiken des digitalen Lesens, nicht nur als Lesen im Digitalen. Hayles schlägt die Unterscheidung von »close«, »hyper« und »machine reading« vor.⁸⁴ Insbesondere letztere integriert das algorithmische Auslesen in den erweiterten Begriff der Alphabetisierung.⁸⁵

Für ein solches Lesen als Prozessieren muss jedoch jeder Code und jedes Programm auf gewisse Weise (ab-)geschlossen sein. Abgeschlossen wird Code im Übergang vom Quelltext zum Programmcode via Compiler. Das verleiht dem Programm Werkcharakter, was bedeutet, dass es nicht mehr ohne Gewaltanwendung umgeschrieben werden kann.⁸⁶ Die Übersetzung von Programmcode in Maschinencode kann wiederum nicht umgekehrt werden. Diese Irreversibilität erlaubt es, den Quellcode mit einem Copyright zu versehen und unzugänglich zu machen. Im Copyright zeigt sich die Notwendigkeit der Schließung des Codes als abgeschlossenes Werk, das wiederum distribuiert und damit auf Computern installiert werden kann. Erst so lassen sich Code/ Programme von Maschinen prozessieren, sodass der/die User:in diese verwenden kann. Der Blick auf die Schriftlichkeit des Codes zeigt, dass diese Schließung selbst nicht Teil der Schrift des Codes ist, sondern seines ideologischen Gebrauchs.

83 Vgl. Hayles: *My Mother*, 59.

84 N. Katherine Hayles: *How we think. Digital Media and Contemporary Technogenesis*. Chicago/London: The Chicago University Press 2012, 55–80.

85 Hayles: *How we think*, 11. Vgl. ferner: Anette Vee. *Code Literacy. How Computer Programming is Changing Writing*. Cambridge MA/London: MIT Press 2017.

86 Umgekehrt stellt die Schriftlichkeit des Codes den Werkcharakter des Programms infrage, worauf Markus Krajewski hinweist. Markus Krajewski: *Against the Power of Algorithms Closing, Literate Programming, and Source Code Critique*. In: *Law Text Culture* 23 (2019), 119–133, hier: 123.

Mit der Geschlossenheit von Programmen geht zudem eine Art Werkcharakter einher, der etwas benötigt, was analogen Texten fremd zu sein scheint. Sie müssen gewartet werden. Damit Programme ausgeführt werden können, bedürfen sie der permanenten Wartung. Durch die Signatur in Bezug auf ihren Quellcode wird dem Programmcode die Notwendigkeit zum Update eingeschrieben, was wiederum die Schließung des Programms aufschiebt, immer weiter in die Zukunft verschiebt und gleichzeitig einen Abschluss erst ermöglicht. Nur die Wartung garantiert das ordnungsgemäße Prozessieren, sprich die Ausführung eines Programms.

Die Schriftlichkeit der Wartung ernst nehmen, bedeutet, nicht nur die Oberfläche des Textes bzw. des Gewebes zu betrachten, nicht nur das Interface oder die Kommandozeile zu analysieren,⁸⁷ sondern auch den »Prozess des Webens« zu berücksichtigen, wie er in technischen Medien der Bild- und Textproduktion wie »Siebdruck, Druckerpressen, Druckmatrizen, Fotografie und Schreibmaschine« zugrunde liegt.⁸⁸

Neben dem Gutenberg-Universum erscheint so noch das Jacquard-Universum des Webstuhls, dessen kurze Geschichte mit Basile Bouchon 1725 beginnt, 1728 von Jean-Baptiste Falcon durch die Lochkarte fortgesetzt, mit Joseph-Marie Jacquard 1801 seinen Namen erhält und von Hermann Hollerith 1884 perfektioniert wird. Und dieses Jacquard-Universum ließe sich wiederum in Phasen einteilen: die *Machinae Arithmeticae* (Wilhelm Schickard), Calculation Engines (Babbage) und dann die Computer.⁸⁹

Das Webstuhl-Universum provoziert das anthropozentrische Denken. Es liefert eine weitere Kränkung des Menschen, da es darauf hinweist, dass menschliches Denken von einer Maschine geleistet werden kann. Zeitgleich zu den Arbeiten von Babbage und Korsakov wird Georg

87 Vgl. dazu Lori Emerson: *Reading Writing Interfaces. From the Digital to the Book-bound*. Minneapolis: University of Minnesota Press 2014.

88 Vgl. Plant: *nullen + einsen*, 76.

89 Vgl. Dotzler: *Papiermaschine*, 43. Aber: »die Machinae Arithmeticae [sind] keine Computer-Vorläufer []; [] keine CONTROL interveniert, keine COMMUNICATION – kein Algorithmus.« (Dotzler: *Papiermaschine*, 49).

Wilhelm Friedrich Hegel mittels der idealistischen Philosophie die Abwehr dieser Mechanisierung des Denkens vorbereiten.⁹⁰

Die beiden medien- und schrifttheoretischen Universen, die sich zeitversetzt ausbilden, stehen nicht in Konkurrenz, sondern sind in ihrer materiellen wie ideellen Konzeption aufeinander bezogen. Anstelle der Software ›Schrift‹, die dem Gutenberg-Universum entstammt, beginnt Ada Lovelace das Jacquard-Universum zu programmieren. Anstelle des Buchdrucks adressiert die Webetechnik⁹¹ eine maschinelle Literatur; anstelle von Papier und beweglichen Lettern definieren Garn und Lochkarten die Materialität digitaler Schriftlichkeit. Daraus ergibt sich das (textile) Prozessieren⁹² als spezifische Lesetechnik.

In Knuths *literate programming* manifestiert sich das Webstuhl-Universum nun im Raum des Programmierens. Den Konnex verdankt dieser Raum dem Namen seiner Programmiersprache: WEB, also Gewebe. Weil dieses Drei-Buchstaben-Wort noch nicht für das Programmieren benutzt wird, gibt es nicht nur den Eigennamen der Sprache, sondern auch das Konzept: WEB, das Gewebe, Netz, das verfilzt oder (ver-)und gewebt werden kann. Damit schreibt sich Knuths *literate programming* in die Geschichte eines Webstuhl-Universums ein. Denn Knuths Programmieren zeichnet sich durch einen doppelten Vorgang aus: durch *weaving* und *tangling*, weben und verfilzen.⁹³ Der ›gewebte‹ Code, den die Maschine ausführt, wird ›verfilzt‹. Zwei Schreib- und Leseoperationen lassen die Tradition des Webstuhls aufleben, von denen alle Programmierer:innen abstammen. *Weaving* und *tangling* sind Schreibmodi, verweisen aber auch auf zwei unterschiedliche Weisen

90 Vgl. Ernst: Schicksal, 41.

91 Vgl. Plant: *nullen + einsen*, 19.

92 Zitiert nach Ernst: Schicksal, 24. Vgl. Birgit Schneider: Diagramm und bildtextile Ordnung. In: *Bildwelten des Wissens*, 3.1 (2005), 9–20.

93 Stiegler hebt die Praxis des Webens auch für die tertiäre Retention hervor. Er schreibt: »This weaving, for which tertiary retention is irreducible, and constitutes noesis insofar as it makes [fait] (the) *différance* at once as exteriorization, reproduction and discernment, is also what undermines [défait] noesis: it is as such that writing is a pharmakon.« Bernard Stiegler: *The Neganthropocene*. O.O.: Open Humanities Press 2018, 229.

des Prozessierens, die von ihren jeweiligen Adressaten:innen abhängen: Mensch und Maschine. An sie ist nicht nur die Wartung der Maschine gekoppelt, sondern auch die des Codes, damit das fortdauernde Funktionieren als Kommunikation gewährleistet bleibt. Im digitalen Kontext von Programmieren, Prozessieren und Codieren ist das Netz nicht Metapher, sondern bestimmt das, was Buchdruck- vom Webstuhl-Universum unterscheidet. Im Gutenberg-Universum ist jedes Zeichen adäquat verräumlicht, während im Webstuhl die Zeichen miteinander verbunden sind.

Beide Universen teilen eine Gemeinsamkeit, nämlich die Konfiguration der Nutzer:innen als Subjekte, die prozessieren, und damit einem Skript folgen müssen, um die Maschinen zu bedienen.⁹⁴ Markus Krajewski und Cornelia Vismann heben den Verwendungszusammenhang des Quellcodes hervor: »Man kann aus ihm mit einem Klick entweder einen Text in Form eines Handbuchs oder einer Dokumentation für andere Entwickler erstellen, oder aber ebenfalls per Knopfdruck eine ausführbare Datei generieren.«⁹⁵

Krajewski spricht von einer »Source Code critique«, die auf den Quellcode als doppelten Text setzt: als Kommentar des Codes, der den Code beinhaltet. Ziel ist die Analyse des »extensive interplay of code and commentary in order to make the algorithms themselves more transparent and thus more comprehensible.«⁹⁶ Die elementaren Praktiken sind dabei Lesen, Kommentieren und Modifizieren: »The source code would thus contain the individual commands and data structures together with their documentation or philological apparatus.«⁹⁷ Quellcode, das gilt es zu beachten, enthält zudem immer mehr als nur den Code: Dort werden auch Kommentare der Programmierer:innen verzeichnet.⁹⁸

94 Shen: GUIDing, 88.

95 Markus Krajewski/Cornelia Vismann: Kommentar, Code und Kodifikation. In: *Zeitschrift für Ideengeschichte* (2009), 5–16, hier: 12.

96 Krajewski: *Against Power*, 128.

97 Krajewski: *Against Power*, 130.

98 Joasia Krysa/Grzesiek Sedek: Source Code. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. Cambridge MA/London: MIT Press, 236–242, hier: 237.

Wendy Hui Kyong Chun ergänzt die Perspektive um einen kritischen Einwand. Für sie ist der Quellcode eine Art Fetisch.⁹⁹ Mit dem Wortspiel »sourcery«, bestehend aus *source* (Quelle) und *sorcery* (Zauberei), verweist Chun einerseits auf die Konstruktion einer Quelle als Ausgangspunkt maschineller Agency als auch auf das Prozessieren als etwas, das »von selbst« geschieht. Demgegenüber konzipiert Chun Code als »re-source«, um erstens den »gap« zwischen Code und Ausführung zu verstehen, zweitens »entropy, noise and decay« von Code zu denken und drittens Code zu historisieren und zu kontextualisieren.¹⁰⁰

Beide methodischen Zugänge (Critical und Source Code Studies) behalten in ihrem Kern ein seit Friedrich Schleiermacher bekanntes Postulat bei: Hermeneutik darf nicht mechanisch, oder in unserem Fall, digital werden, damit sich ein Verstehen als aktives Ereignis einstellt.¹⁰¹

Krajweski bezieht sich für seine Praxis der Source Code Critique auf Knuths *The Art of Computer Programming*. Knuth konzipiert ein *literate programming*, »das sich als dichte Verbindung von algorithmischen Strukturen und ihren Kommentaren zeigt.«¹⁰² Das doppelte Schreiben von Weben und Verfilzen in Knuths *literate programming* inklusive des ökonomischen Kalküls seiner Korrigierbarkeit ermöglicht nämlich »style«. Und dieser entwickelt sich.¹⁰³ Programmieren in WEB wird zu einer artistischen Angelegenheit, die dem Programmieren wieder einen geniehaften, auktorialen Pinselstrich zurückgibt. Hier schreibt sich explizit der/die Programmierer:in als Autor:in ein. Eine alte literarische

99 Wendy Hui Kyong Chun: On »Sourcery«, or Code as Fetish. In: *Configurations* 16/3 (2008), 299–324.

100 Chun: *Sourcery*, 321f.

101 Vgl. Dotzler: *Papiermaschine*, 16: »Die Auslegung wird hier zur Rechnung, die Hermeneutik, indem sie die lebendige Operation darstellen will, zum puren Mechanismus«, wird Dilthey mit Rücksicht auf jene Rückschau, die ihm den Blick eingesenkt hat, kritisieren, um dagegen an eben jene neue Position zu erinnern, wie Schleiermacher sie definiert hat: »daß sich das Mißverstehen von selbst ergibt und das Verstehen auf jedem Punkt muß gewollt und gesucht werden«.

102 Krajewski/Vismann: Kommentar, 12.

103 Knuth: *Literate Programming*, 108.

Autor:innen-Funktion wird in das Konzept des Programmierens eingeschmuggelt. Dieser Schmuggel erlaubt es aber auch, eine Vorschrift für Autor:innenschaft zu setzen: »you have the freedom to say anything you want, and this freedom entails the responsibility of deciding what to say.«¹⁰⁴ Daraus resultiert, dass die Programme weniger Fehler aufweisen, weil ein starkes Autor:innensubjekt gezwungen ist, »to clarify my thoughts as I do the programming.« Das große »I« ist dabei schwerer zu täuschen.¹⁰⁵ Der Schmuggel hat also einen Hang zur Wahrheit.

Das Ich mit großem I (wie es auch Knuths *literate programming* voraussetzt) hat aber nicht immer die absolute Kontrolle. Die Les- bzw. Prozessierbarkeit des Werks ist stets bedroht, und zwar von ganz unscheinbaren nicht-digitalen Entitäten. So hat eine Motte schon einmal eine Rechenmaschine lahmgelegt.¹⁰⁶ Grace Hopper entdeckte beim Versuch, das Problem der Maschine zu lösen, den ersten »bug«. Seit diesem Vorfall werden Computer und ihre Netzwerke von Insekten bevölkert. Die Schrift kehrt über den »bug« (den Parasiten) als Öffnung wieder in die abgeschlossene Tätigkeit des (Programm- bzw. Code-)Prozessierens ein. Das »De-Bugging« wird im Kontext der digitalen Schriftlichkeit zu einem entscheidenden Einstiegspunkt der kritischen Auseinandersetzung mit Code als Schrift und der Materialität der Maschine, weil Prozessierung, also Leseprozesse unterbrochen und neu gestartet werden können und müssen.

3. Codieren: Vom einzelnen Zeichen zum Algorithmus

Erst als Programme auch als abgeschlossene Texte in höheren Programmiersprachen geschrieben wurden und Inhalte bereitstellten, konnte Codieren wiederum einen Teilaspekt des Schreibprozesses bezeichnen. Code ist von Rede und Schrift dadurch unterschieden, »in that it exists in clearly differentiated versions that are executable in a process that

104 Knuth: *Literate Programming*, 108.

105 Knuth: *Literate Programming*, 108.

106 Plant: *nullen + einsen*, 134.

includes hardware and software and that makes obsolete programs literally unplayable unless an emulator or archival machine is used.«¹⁰⁷ Das bedeutet auch, dass Codes Bausteine eines Programms sind wie Sätze in einem Text. Code ist hierarchisiert: C++ zum Beispiel ist in ASCII geschrieben, wobei der ASCII-Code wiederum in Maschinensprache übersetzt wird.¹⁰⁸

Im Grunde meint Codieren, ein Zeichen mittels eines Codes in ein anderes Zeichen zu verwandeln. Dieser Prozess wird allerdings schnell standardisiert, sodass sich Baustein-codes ergeben, die wiederum zum Programmieren verwendet werden können. Code ist in der meisten Software, die wir im Alltag benutzen, ›alter‹ Code, der um neue Bestandteile ergänzt wird,¹⁰⁹ ganz so wie bestimmte Elemente in Narrativen immer wieder benutzt werden, um etwas verständlich auszudrücken oder damit ein Text ›funktioniert‹. Die Assoziation liegt nahe, hier von Intertextualität des Codes zu sprechen. Doch nicht allein anhand seiner Bausteine kann der Code erkannt werden. Friedrich Kittlers Skepsis, wir könnten »schlichtweg nicht mehr wissen, was unser Schreiben tut, und beim Programmieren am allerwenigsten«,¹¹⁰ muss nicht geteilt werden. Durchaus kann mittels Wartungsarbeit beschrieben werden, was die Schrift respektive der Code tun, mithin was digitale Schriftlichkeit tut. Es geht nicht mehr um den Schreibprozess, sondern um einen Schrift- oder Codeprozess, der von Menschen und

107 Hayles: *My Mother*, 52.

108 Hayles: *My Mother*, 57. Damit unterscheidet sich C++ von Programmiersprachen wie FORTRAN oder BASIC. Zu unterscheiden sind ferner prozedurale von objekt-orientierten Programmierungen (Hayles: *My Mother*, 57), die entsprechend der Syntax natürlicher Sprache konzipiert sind. Prozedurale Programmsprachen dagegen verwenden sogenanntes »early binding«, verbinden also Programm, Compiler und Maschine bevor das Programm eigentlich läuft (Hayles: *My Mother*, 59).

109 Marisa Leavitt Cohn: Keeping Software present: Software as a Timely Object for STS Studies of the Digital. In: Janet Vertesi/David Ribes: *digitalSTS. A Field Guide for Science & Technology Studies*. Princeton UP 2019, 423–446.

110 Friedrich Kittler: Es gibt keine Software. In: ders.: *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam 1993, 225–242, hier: 229.

Maschinen gelesen werden kann. Das Geschriebene gewährleistet das Anschreiben, das Adressieren meint, und adressiert wird die Software genauso wie die Hardware. Um zu erfahren, was nun der Code tut, bedarf es gewisser Vorsichtsmaßnahmen. Erstens stellt die Adressierung der Software eine Souveränitätssimulation dar, die das eigentliche Schreiben des Codes nicht mehr betrifft. Die digitale Schriftlichkeit untersucht und verkörpert zugleich die Amalgamierung von Soft- und Hardware, jene Reduktion der Schrift auf »Maschinenzwänge«,¹¹¹ wie sie in jedem Compiler stattfindet. Entscheidend nämlich ist, inwiefern Hardware Heimat eines Schreibsystems sein kann.¹¹² Folglich ist eine »Mineralogie des Geistes«¹¹³ nur mit Computern zu beschreiben. Silizium und Siliziumoxid und noch viele andere seltenen Erden führt Jussi Parikka in einer Geologie der Medien zusammen.¹¹⁴

Zweitens also muss die genuine Materialität des Codes berücksichtigt werden. Schriftlichkeit hat eine offensichtliche materielle Dimension: Papier, Tinte, Schreibgeräte. Und dennoch ist gerade der Schriftlichkeit das Prozessuale und Emergente von Schrift und Schreiben eingetragen, sodass auch digitale Schriftlichkeit eine spezifische Verbindung zu Prozess und Materialität unterhält: Computer, Siliziumchips, Serverfarmen, Plastik, seltene Erden etc. auf der einen Seite, ein manchmal auch KI-gestütztes Produzieren von Text auf der anderen Seite, das auch seine neuronale Erinnerungsspur kennt: das Engramm.¹¹⁵ Im analogen wie

111 Kittler: Es gibt keine Software, 237.

112 Kittler: Es gibt keine Software, 239.

113 Emanuele Coccia: *Das Gute in den Dingen*. Berlin: Merve 2017, 17.

114 Jussi Parikka: *A Geology of Media*. Minneapolis: University of Minnesota Press 2015. Vgl. auch Sarah Pink/Elisanda Ardëvol/Déborá Lanzeni: Digital Materiality. In: dies.: *Digital Materialities. Design and Anthropology*. London/New York: Routledge 2016, 1–26, hier: 10, die im Kontext der Begriffe Design, Anthropologie und Umwelt die Materialität des Digitalen nicht als *a priori* gegeben sehen oder sie als Endpunkt einer Analyse begreifen, sondern »as a process, and as emergent«.

115 Vgl. zu den Engrammen Shintaro Miyazaki: Medien-Archäo-Graffiti. In: Moritz Hiller/Stefan Höltgen (Hg.): *Archäographien. Aspekte einer radikalen Medienarchäologie*. Schwabe Verlag: Basel 2019, 13–20, hier: 14.

digitalen Fall, die sich nur graduell unterscheiden und sich historisch verorten lassen, benötigt eine Grammatologie des Codes auch eine Betrachtung der Materialität.¹¹⁶ Eine digitale Grammatologie ist nicht ohne ihre Hardware zu denken. Codes, Programme und Software werden so zu digitalen Objekten, wie der Technikphilosoph Yuk Hui argumentiert.¹¹⁷

Drittens gilt es, die Gesten und Praktiken des Codes zu analysieren. Im Feld der Science & Technology Studies wird zunehmend Software als soziale Praxis¹¹⁸ in den Blick genommen und damit in das Paradigma der Kommunikation und Interaktion eingeschrieben: »Software as a whole is not only ›code‹ but a symbolic form involving cultural practices of its employment and appropriation.«¹¹⁹ Carl DiSalvo bestimmt, wie Software funktioniert, wie folgt, nämlich »through associations, interactions, and performances between and among individuals, groups, organizations and code.«¹²⁰ So zeigt sich eine spezifische Affordanz der Software, die von ihrer Codierung abhängig ist. Marissa Leavitt Cohn verortet Software »somewhere in the gray middle, somewhere between the excess of code's present performance, and the inevitable decay and obsolescence of software diagrams, documentation, and representations.«¹²¹ Cohn gibt folglich das Diktum aus, Software ›präsenz‹ zu halten. Diese Vorstellung des ständigen Updatens von Software setzt nicht nur Übersetzungs- und Interpretationsprozesse in Gang, sondern

116 Vgl. Laura Forlano: Materiality. Introduction. In: Janet Vertesi/David Ribes: *digitalSTS. A Field Guide for Science & Technology Studies*. Princeton UP 2019, 11–16, hier: 13.

117 Yuk Hui: *On the Existence of Digital Objects*. Minneapolis: University of Minnesota Press 2016.

118 Carl DiSalvo: Software. Introduction. In: Janet Vertesi/David Ribes: *digitalSTS. A Field Guide for Science & Technology Studies*. Princeton UP 2019, 365–368, hier 365.

119 Cramer: Language, 173. Und weiter heißt es: »But since writing in a computer control language is what materially makes up software, critical thinking about computers is not possible without an informed understanding of the structural formalism of its control languages.«

120 DiSalvo: Introduction, 365.

121 Cohn: Keeping Software present, 425.

stellt ins Zentrum, was Kittler die Korrigierbarkeit der Schrift nennt: »Die Korrektur von Geschriebenem scheint zunächst ein subjektiv willkürlicher Eingriff in den bereits geschriebenen Text. Aber sie gehorcht doch zugleich dem Text; sie tilgt, was nicht in ihn hinein gehört.«¹²² Hier schließen wir wieder an die Wartungsarbeit als Sorge der digitalen Schriftlichkeit an, die dann auch eine politische Dimension erreicht, sobald das Codieren automatisiert wird.

Hatten wir bereits davon gesprochen, dass Codieren das Umformen von Zeichen meint, so lässt sich auch diese Prozedur automatisieren. Auf diese Weise ergeben sich Algorithmen. Algorithmus (eigentlich Algorismus),¹²³ die Latinisierung des Eigennamens des Universalgelehrten Abu Dscha'far Muhammad ibn Musa al-Chwārizmī,¹²⁴ der ein Buch über den Gebrauch indischer Zahlzeichen schrieb,¹²⁵ bezeichnet »[e]ine Regel, die erlaubt, eine Operation, die ausschließlich mit Zeichen arbeitet, als eine effektive Prozedur zu realisieren.«¹²⁶ ALGORITHM = LOGIC + CONTROL.¹²⁷ Die Formel bezeichnet, so Dotzler, dass jeder Algorithmus auf seine Effizienz hin spezifiziert werden kann und muss. Was zur Folge hat, dass es »keinen Algorithmus [gibt] ohne Befehlsgewalt.«¹²⁸

Mit der Turingmaschine rücken Algorithmen ins Zentrum der »computing science«.¹²⁹ Algorithmen gehören der »pragmatic dimen-

122 Friedrich Kittler: *Baggersee. Frühe Schriften aus dem Nachlass*. Paderborn: Fink 2015, 168.

123 Vgl. Dotzler: *Papiermaschine*, 69.

124 Krajewski: *Against Power*, 122 und: Friedrich Kittler: *Die Endlichkeit der Algorithmen*: <https://archive.transmediale.de/de/content/die-endlichkeit-der-algorithmen> (zuletzt abgerufen am 23.11.2023).

125 Dotzler: *Papiermaschine*, 69.

126 Sybille Krämer: *Wieso gilt Ada Lovelace als die »erste Programmiererin« und was bedeutet überhaupt »programmieren«?* In: dies. (Hg.): *Ada Lovelace: Die Pionierin der Computertechnik und ihre Nachfolgerinnen*. München: Fink 2015, 75–90, hier: 79.

127 Dotzler: *Papiermaschine*, 8: Die Formel geht auf Robert Kowalski zurück.

128 Dotzler: *Papiermaschine*, 8.

129 Andrew Goffey: *Algorithm*. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. Cambridge MA/London: MIT Press, 15–20, hier: 16.

sion of programming« an, weil sie etwas tun.¹³⁰ Ihre Agentialität ist entscheidend. Mit anderen Worten: Algorithmen sind Sprechakte. Andrew Goffey begreift Algorithmus sogar als ›Aussage‹ im Sinne Michel Foucaults.¹³¹ Während Foucault Aussagen an Subjekte koppelt, argumentiert Goffey, dass Algorithmen als Aussagen einen transversalen Maschinendiskurs konfigurieren, in denen Menschen und Maschinen als Subjekte artikuliert werden und aufeinander einwirken.¹³²

Sybille Krämer bestimmt Algorithmen ferner anhand von vier Merkmalen: Allgemeingültigkeit, Finitheit, Eindeutigkeit, Variabilität.¹³³ Sie weist darauf hin, dass ein Algorithmus noch kein Programm sei: »Algorithmen sind das Rohmaterial für die Programmierung, das also, was die Schnürsenkel beim Schuhezubinden und was die Backzutaten für das Backen sind.«¹³⁴ Kittler definiert Algorithmen mit drei Begriffen: »Determiniertheit«, gleiche Bedingungen liefern immer wieder denselben Output, »Determinismus«, die Abfolge der Schritte ist immer gleich, und »Terminierung«, der »Algorithmus hält für jede Eingabe nach endlich vielen Schritten an.«¹³⁵ Glaubt man Kittler, hat Leibniz den Algorithmus eingeführt: »Leibniz hat 1684 den Calculus, die Differentialrechnung, ganz formal in ihren Grundregeln beschrieben und das dann Algorithmus genannt.«¹³⁶ Krämer präzisiert mit Bezug auf Leibniz und definiert Kalkül:

»Ein Kalkül ist eine beschränkte Menge von Zeichen in Form eines Alphabets, Formationsregeln geben an, wie die Grundzeichen zu Ausdrücken (*expressiones*) aneinander gefügt werden, und Transformati-

130 Goffey: Algorithm, 16f.

131 Goffey: Algorithm, 17.

132 Goffey: Algorithm, 19.

133 Krämer: Wieso gilt, 77f.

134 Krämer: Wieso gilt, 79.

135 Kittler: <https://archive.transmediale.de/de/content/die-endlichkeit-der-algorithmen> (zuletzt abgerufen am 23.11.2023).

136 Kittler: <https://archive.transmediale.de/de/content/die-endlichkeit-der-algorithmen> (zuletzt abgerufen am 23.11.2023).

onsregeln stellen Beziehungen zwischen den Ausdrücken her, damit diese ineinander umgeformt werden können.«¹³⁷

Nach dem klassisch zu nennenden Handbuch *Introduction to Algorithms* ist der Algorithmus eine Funktion, die Werte nimmt (Input) und Werte ausgibt (Output).¹³⁸ Daraus resultieren zwei Fragen: Erstens, welche Input-Daten werden verwendet, und zweitens, wie funktioniert der Algorithmus? Beide Fragen umkreisen das Feld der digitalen Schriftlichkeit und ihrer Bedeutung für eine kritische Theorie der Mensch-Code-Schrift-Beziehung. Es ist nicht so wichtig, ob die *Blackbox* »Algorithmus« geöffnet wird. Denn wichtiger ist, zu verstehen, welchen Input der Algorithmus bekommt und woher der Input kommt.¹³⁹

Algorithmen sind vor allem »networked information algorithm«.¹⁴⁰ Damit ist nicht nur der Algorithmus in der Vorstellung der Informatik gemeint, als Methode mit Input bestimmten Outcome zu generieren. Vielmehr wird der Algorithmus als Bestandteil eines Akteur-Netzwerks mit dem Menschen aufgefasst. Zu diesem Netzwerk gehören dann nicht nur Menschen und Maschinen, sondern normierte und institutionalisierte Algorithmen und Codes, menschliche Praktiken ihrer Anwendung sowie die daraus wiederverwendbaren Daten über die Nutzer:innen.¹⁴¹

Für die digitale Schriftlichkeit ist die Schrift nicht mehr nur wie bei Leibniz binär, also ausgehend vom chinesischen I Ching als »ideographic writing«¹⁴² konzipiert und damit nicht mehr durch 1 und 0 bestimmt,

137 Krämer: Wieso gilt, 81.

138 Thomas H. Cormen/Charles E. Leiserson/Ronald L. Rivest/Clifford Stein: *Introduction to Algorithms*. Cambridge, MA: The MIT Press 4 2022.

139 Mike Ananny: Toward an Ethics of Algorithms: Convening, Observation, Probability, and Timeliness. In: *Science, Technology, & Human Values* 4/1 (2016), 93–117, hier: 98.

140 Ananny: Ethics, 97.

141 Ananny: Ethics, 98.

142 Hui: *Digital Objects*, 16. Zu einer grundlegenden Kritik des Binären im Digitalen siehe: Aden Evens: *The Digital and Its Discontents*. Minneapolis: University of Minnesota Press 2024.

sondern »as the capacity to process data.«¹⁴³ Und auch Yuk Hui erkennt, dass Algorithmen an unseren Gedanken mitschreiben. Der Mensch ist dabei gestellt, im Sinne von Martin Heideggers Begriff des *Gestells*, in ein *Semantic Web* nach Tim Berners-Lee oder in Edgar Todds *relational database*. Damit ermöglichen Algorithmen eine tertiäre Protention der Schrift.¹⁴⁴ Hat Bernard Stiegler in Bezug auf Derridas Kritik an Husserl und den phänomenologischen Konzepten von primärer und sekundärer Retention eine dritte Form herausgearbeitet, wendet Yuk Hui dieses Ergebnis nun auch auf die Protention an, wodurch mit »digital writing, synchronization has become much more powerful, becoming the very foundation of what we call digital cultures.«¹⁴⁵

Und weil Algorithmen in einem Beziehungsgeflecht zu ihren *User:innen* stehen, wie bewusst dieses Verhältnis auch immer sein mag, verändern sie sich mit ihrer Benutzung.¹⁴⁶ Es kommt zu einer Personalisierung, einer Kooperation zwischen Mensch und Maschine, Input und Output.¹⁴⁷ Das heißt, Benutzen und Schreiben fällt hier in eins. Aber auch das digitale Archiv spielt bei der Konfiguration von Subjekten und einer digitalen Kultur, wie Wolfgang Ernst analysiert hat, eine tragende Rolle.¹⁴⁸

Algorithmen, die aus in sich geschlossenen Befehlsketten bestehen,¹⁴⁹ können als Code bezeichnet werden. Damit dieser Code ausgeführt werden kann, bedarf es erneut eines Körpers,¹⁵⁰ sprich ei-

143 Hui: *Digital Objects*, 25.

144 Hui: *Digital Objects*, 38.

145 Hui: *Writing*, 22.

146 Vgl. Dazu: Ulises Ali Mejias: *Off the Network. Disrupting the Digital World*. Minneapolis: University of Minnesota Press 2013.

147 Nick Seaver: *Knowing Algorithms*. In: Janet Vertesi/David Ribes: *digitalSTS. A Field Guide for Science & Technology Studies*. Princeton UP 2019, 412–422, hier: 415 und 419.

148 Vgl. Wolfgang Ernst: *Digital Memory and the Archive*. Minneapolis: University of Minnesota Press 2012. Wolfgang Ernst: *Das Rumoren der Archive*. Berlin: Merve 2002, 129–140.

149 Krajewski: *Against the Power*, 122.

150 Krajewski: *Against the Power*, 122.

nes Computers. Die Ausführbarkeit verändert aber den Algorithmus, wodurch der Schreibprozess nicht mehr an ein intentionales Autor:inensubjekt gekoppelt ist, sondern sich als ein Akteur-Netzwerk nicht-menschlicher und menschlicher *User:innen* erweist. Problematisch wird diese Beziehung erst dann, wenn es, wie Stiegler und McKenzie Wark kritisieren, asymmetrisch und intransparent wird.¹⁵¹ Dann gilt tatsächlich jene erste Erwähnung in lateinischer Sprache, die den Algorithmus als Herrschaftszeichen einführt: »*Dixit Algorithmi: laudes deo rectori nostro*«. ¹⁵²

Gesetz und Befehl sind dem bzw. der Programmierer:in und dem bzw. der Coder:in nicht fremd, ganz im Gegenteil: »Code is law«. ¹⁵³ Sie arbeiten am Algorithmus, der eine spezifische Ausführbarkeit gewährleistet. Der Umweg, der für eine andere Interpretation nötig ist, geht über eine Diskursanalyse, die »nach dem Algorithmus sucht«, wie Philipp Sarasin meint, d.h. nach dem Programmcode für eine spezifisch wiederkehrende Folge von Aussagen bzw. für die Möglichkeitsbedingungen von Aussagen und ihrer Ausschlussmechanismen.

Und weil *User:innen* »für Regierungen wie für Medienindustrien grundsätzlich nicht vertrauenswürdig [sind] und von den Potentialen fortgeschrittener Technik ferngehalten werden [müssen]«, ¹⁵⁴ wird es umso erforderlicher, dass sich die Geisteswissenschaften mit den Skripten, dem Codieren, Programmieren und Prozessieren nicht nur von Computern, sondern auch der »Wetware« Mensch beschäftigen. Denn wir interpretieren zwar schon immer Daten, aber seit der Aufklärung – dem Befehl zum Selbstdenken – selten Befehle.

151 Stiegler: *Neganthropocene* und McKenzie Wark: *Das Kapital ist tot*. Berlin: Merve 2021.

152 Kittler: Endlichkeit: <https://archive.transmediale.de/de/content/die-endliche-it-der-algorithmen> (zuletzt abgerufen am 15.11.2023): »Es sagt der Algorithmus gelobt sei Gott, unser Herr«.

153 Goffey: *Algorithm*, 19.

154 Heilmann: *Textverarbeitung*, 45.

Anstelle von Masterdiskursen müsste im 21. Jahrhundert die Rede von Masteralgorithmen sein,¹⁵⁵ die unseren Alltag aus Überwachung, Nudging, Social Media etc. prägen. Es wird dabei zur Aufgabe, diese Abhängigkeit, wie Nick Seaver erklärt, im Modus der Anagnorisis (wieder) zu erkennen¹⁵⁶ oder zu reparieren.

Eine Reparaturpolitik gegenüber dem Algorithmus fordern Julia Velkova und Anne Kaun. Sie hat das Austricksen von Algorithmen im Blick, rekurriert also einerseits auf die algorithmische Anagnorisis Seavers als auch auf die kapitalismuskritischen Einwürfe von Stiegler und Wark. Was sich abzeichnet, ist ein diskursiver Bruch in der Bedeutung von Algorithmen. Waren Algorithmen einst Bestandteile von Programmen, sind sie heute zunehmend eigenständige Akteure, die wie Quasi-Programme, Datenverarbeitung im Digitalen steuern und strukturieren. Algorithmen werden zu machtvollen Objekten stilisiert, auch durch kritische Forschung, die ihre politische, sozio-ökonomische und ästhetische Bedeutung betonen, interpretieren und analysieren.¹⁵⁷ Die digitale Schriftlichkeit der Codes zu analysieren heißt auch, nach dem »Verhalten«¹⁵⁸ von Schrift, spezieller von Code und Algorithmen zu fragen.

Der Zusammenhang von Stimme und Schrift wird auch im Kontext von Coding wichtig, da Algorithmen Stimmen »verstärken oder unterdrücken«.¹⁵⁹ Stimme ist im weiteren Sinne gemeint, im Sinne von »eine Stimme haben« oder »sich Gehör verschaffen«. Algorithmen haben also einen »material effect on end users«.¹⁶⁰ Dagegen bleibt das Coding, die

155 Vgl. Pedro Dominges: *The Master Algorithm. How the Quest for the Ultimate Learning Machine Will Remake Our World*. New York: Basic Books 2015.

156 Seaver: *Knowing Algorithms*, 414f.

157 Julia Velkova/Anne Kaun: Algorithmischer Widerstand. Medienpraxis und Reparaturpolitik. In: Daniel Neugebauer (Hg.): *Haut und Code. Publikationsreihe Das Neue Alphabet*. Band 5. Leipzig: Spector Books 2021, 59–82, hier: 63f. Vgl. auch weiterführend: Seb Franklin: *The Digitally Disposed. Racial Capitalism and the Informatics of Value*. Minneapolis: University of Minnesota Press 2021.

158 Velkova/Kaun: *Algorithmischer Widerstand*, 65.

159 Velkova/Kaun: *Algorithmischer Widerstand*, 66.

160 Coffey: *Algorithm*, 15.

Schrift des Algorithmus sekundär, als Hilfsmittel diese Stimme zu verstärken oder abzuschwächen.

Die Reparaturpolitik zielt darauf ab, die programmierten Algorithmen durch ein Andersnutzen in neue Bahnen zu lenken, sodass Rückwärtsbildersuchen, überhaupt Suchanfragen etc. diversere Ergebnisse zeitigen. Dabei geht es nicht darum, in die Schriftlichkeit des Codes einzugreifen und diesen umzuschreiben, sondern vielmehr in der Souveränitätssimulation Interface-Anwendungstaktiken zu entwickeln, die die Nutzung verbessern.¹⁶¹ »Reparatur« ist hier eine Metapher, die für den symbolischen Akt der Korrektur einer empfundenen ›Störung‹ eines algorithmischen Systems steht, womit auch die dominante Position des Systems selbst in Frage gestellt werden kann.¹⁶² Die Reparaturpolitik ist zudem an die Praxis des Einschreibens als Hacking gekoppelt,¹⁶³ gerade dort, wo die »Repair-Initiativen« im Sinne des Technikphilosophen Gilbert Simondon »vom unfertigen und verletzten Objekt her« gedacht werden.¹⁶⁴

Folglich findet sich bereits bei Ada Lovelace nicht nur eine Ökonomie des Kalküls, wenn sie von der Optimierung des Codes spricht,¹⁶⁵ sondern auch eine potenzielle Reparierbar- und Korrigierbarkeit, die der Optimierung immanent ist. Zwar kann Optimierung im neoliberal-kapitalistischen Sinne als Effizienzmaximierung verstanden werden, und vor

161 Vgl. Velkova/Kaun: Algorithmischer Widerstand, 79.

162 Velkova/Kaun: Algorithmischer Widerstand, 80.

163 Zum Hacking vgl. u.a. McKenzie Wark: *Hacker Manifesto. A Hacker Manifesto*. Aus dem Englischen von Dietmar Zimmer. München: C.H. Beck 2005 und Sophie Toupin: Feministisches Hacking. Widerstand durch das Schaffen neuer Räume. In: Cornelia Sollfrank (Hg.): *Die schönen Kriegerinnen. Technofeministische Praxis im 21. Jahrhundert*. Wien: Transversal texts 2018, S. 33–58.

164 Barbara Eder: *Das Denken der Maschine. Marx, Mumford, Simondon*. Wien/Berlin: Mandelbaum 2023, 55.

165 Vgl. Bernhard J. Dotzler: Anmerkungen der Übersetzerin. Charles Babbage und Ada Augusta Lovelace in Kooperation. In: Sybille Krämer (Hg.): *Ada Lovelace. Die Pionierin der Computertechnik und ihre Nachfolgerinnen*. Paderborn: Fink 2015, 53–68, hier: 65.

allem geht es um eine Ökonomie des Kalküls, wenn die Zeit zur Berechnung minimiert werden soll, aber allein mit der Manipulation des Codes, der nicht von sich aus bereits ›perfekt‹ ist, eröffnet sich eine Möglichkeit des Eingreifens und des Widerstands. Und dieses Moment ist für die digitale Schriftlichkeit konstitutiv, weil sich hier ihre dekonstruktive Haltung erneuert.

In dieser Perspektive wird deutlich, welche Rolle eine digitale Schriftlichkeit spielen kann, und zwar in Bezug auf eine kritische (Quell-)Codeanalyse in soziopolitischen und kapitalistischen Zusammenhängen. Historisch lässt sich die generalisierende Amalgamierung von Schriftlichkeit und Digitalität als Grammatisierung mit Beginn des *World Wide Web* datieren: 1993.¹⁶⁶ Wie schon bei den ›networked information algorithms‹ geht es um das Zusammenwirken von nicht-menschlichen und menschlichen Akteur:innen. Mit Stiegler lassen sich die Schreibweisen näher bestimmen und unterscheiden.¹⁶⁷ Als erste Schreibweise kann das Codieren von ›networked information algorithms‹ bezeichnet werden, die von ihrer Benutzung ständig weiterentwickelt, mithin ›gefüttert‹ werden. Dieses Codieren ordnet alle menschliche Tätigkeit dem Paradigma des *content creating* unter,¹⁶⁸ das auch Wark analysiert.¹⁶⁹ Durch die digitale Grammatisierung entstehen neue Klassen, wie die »Vektorialistenklasse«, die den Vektor der Informationsflüsse besitzt und kontrolliert, und einem »Kognitariat«, das neue Information produziert. Dazu zählen Schriftsteller:innen ebenso wie Wissenschaftler:innen sowie viele andere mehr, die täglich etwas im Internet posten, verbreiten oder auf andere Art und Weise vernetzt sind.¹⁷⁰ Die andere Schreibweise bezeichnet ein Codieren als Produzieren von sogenannter ›free software‹, die nicht kapitalisiert und

166 Stiegler: *Neganthropocene*, 144.

167 Stiegler: *Neganthropocene*, 144.

168 Stiegler: *Neganthropocene*, 144.

169 Vgl. Wark: *Das Kapital*.

170 Wark: *Das Kapital*, 69.

damit nicht mit einem bzw. einer spezifischen Autor:in und als Werk mit einem urheberrechtlich geschützten Produkt einhergehen.¹⁷¹

Weil aber die erste Schreibweise der digitalen Grammmatisierung die dominierende ist, entstehen so weitere prekäre Arbeitsfelder wie das der Commercial Content Moderators (CCMs), die »im Auftrag von externen Social-Media-Zulieferern und Drittanbietern an der Illusion einer unbedarften Medienpräsentation« arbeiten.¹⁷² Diese CCMs durchsuchen die einzelnen sozialen Plattformen nach Bildern und Beiträgen, deren Inhalte »Bombenanschläge und Enthauptungen, Gewalt im Drogenkrieg, Spam, pornografische Bilder aus globalen Sexindustrien, Selbstmordnachrichten und Hilferufe von allen erdenklichen Orten der Welt« zeigen.¹⁷³ Diese »Reinigungsarbeit« (im Gegensatz zur Politik der Reparaturarbeit) wird gerade nicht von Algorithmen übernommen. Sie zeigt nicht nur, wie sehr der Mensch als Arbeitskraft ausgebeutet wird, sondern erinnert einmal mehr an den materiellen Aufwand der scheinbar immateriellen digitalen Kommunikation.

4. Die Beiträge

Gabriele Gramelsberger macht in ihrem Beitrag *Von ›bits‹ zu ›words‹. John von Neumanns linguistic turn als Ursprung digitaler Schriftlichkeit* die Bedeutung der Schriftlichkeit für die Maschinensprache deutlich. Mittels der Transformationsschübe Mechanisierung und Elektrifizierung wird Sprache für Maschinen gebräuchlich. John von Neumann ist es, der Maschinencodes insofern normierte, als er ihre Sprache als Microcode vereinheitlichte. Diese »mnemotechnische Revolution des Maschinencodes«, so Gramelsberger, erlaubt es, Maschinen und Menschen sprachlich zu vernetzen. Das Ergebnis sind »autooperative Schriften«, die einen deklarativen Zeichengebrauch inaugrieren. Dadurch werden die protodigitalen Bedingungen der digitalen Schriftlichkeit sichtbar,

171 Stiegler: *Neganthropocene*, 144.

172 Eder: *Das Denken*, 17.

173 Eder: *Das Denken*, 17.

die sich aus dem *linguistic turn* von Neumanns ableiten. Darin liegt allerdings ein Problem: Das »a-word-at-a-time-thinking« des *linguistic turns* führt zu einer Fülle an Daten, die nicht mehr vom anthropomorphen Schriftgebrauch des Codes bewältigt werden können. Die digitale Revolution geht demnach mit einer Deanthropomorphisierung des Digitalen einher.

Tilo Reifenstein unterzieht in seinem Beitrag (*Elektrographisches Schreiben als Praxis in Bild, Schrift und Material*) das Schrift-Bildlichkeitsparadigma anhand der digitalen Schriftlichkeit einer kritischen Revision. Dabei betont er eine nicht nur medien-, sondern auch eine materialspezifische Betrachtungsweise der digitalen Schriftlichkeit. Im Rückgriff auf zeitgenössische kulturwissenschaftliche Theorien zur Schrift im Digitalen verortet Reifenstein diese gerade in der Bildmaterialität: ASCII, Unicode, und Optical Character Recognition (OCR)-Verfahren sind damit – ebenso wie digitale Schriftlichkeit – nicht ohne die Materialität einer Bild-Schriftlichkeit zu denken. Dabei adaptiert der Beitrag nicht einfach die Humboldt'schen Begriffe, sondern stellt gerade ihre anthropozentrische Perspektive heraus, um dagegen einen prozessontologischen Begriff digitaler Schriftlichkeit zu gewinnen.

Wilhelm von Humboldt steht im Fokus von Lucas Falkenhains Beitrag *Maschinens Schreiben zwischen Energeia und Ergon. Eine sprachphilosophische Untersuchung von Large Language Models*. Falkenhain bezieht dabei Humboldts Unterscheidung von Energeia (Tätigkeit) und Ergon (Werk) auf die Mensch-Computer-Kommunikation, um die Agentialität maschinellen Schreibens zu erfassen. Ferner zeigt der Beitrag, dass mit Humboldts Konzept der Weltlichkeit algorithmische Sprachmodelle auf die Probe gestellt werden können. Das hat Konsequenzen nicht nur für das Maschinensprechen, sondern auch für die Kognitionswissenschaft und die Frage nach einer künstlichen Intelligenz.

Im Zentrum von Julia Nantkes Beitrag *Schreiben und Lesen als Mensch-Maschine-Kommunikation* steht eben jene titelgebende Relation von Mensch und Maschine. Ausgehend von Umberto Ecos Kommunikationsmodell, das auf die Passgenauigkeit und Kenntnis des verwendeten Codes verweist, basiert eine funktionierende Mensch-Maschine-Kommunikation auch auf kontinuierlichen Übersetzungen. Nantke differen-

ziert ihrerseits diese Konstellation aus: In Bezug auf Optical Character Recognition (OCR) lassen sich so Lesen und Schreiben als Decodierung begreifen. Ferner muss auch die Frage nach der Vorgabe des Zielcodes und seinem Autor:innensubjekt gestellt werden. KI-Trainingsprozesse sind dahingehend von den Trainer:innen und ihren ethischen Normen abhängig. Aber auch umgekehrt kann eine nicht-geleitete KI Muster erkennen, um ihre ›Botschaft‹ zu adressieren. Generative Algorithmen und maschinelles Lernen sind aber drittens nicht nur auf plausible Alltagskommunikation beschränkt. Vielmehr lässt sich mit Eco auch, wie Nantke argumentiert, die Maschine als ästhetische Codespenderin begreifen.

In seinem Beitrag *Romantische Maschinen oder: Ein Bericht für ein Literaturhaus* geht Philipp Schönthaler auf das Spannungsverhältnis von Computerliteratur und Romantik ein. Er zeigt, dass sich in der Gegenwartsliteratur – zum Beispiel in Daniel Kehlmanns *Mein Algorithmus und ich* – ein neues Verhältnis zu schreibenden Maschinen abzeichnet, das gar nicht so neu ist: Computer sind nicht mehr rationale, sondern romantische Schreibmaschinen. In der Gegenüberstellung des zeitgenössischen Diskurses mit der Romantik und der Computerliteratur der 1960er Jahre um Max Bense zeigt sich so eine Re-Romantisierung von Poesie, Werk und Autorschaft. Die Rationalität der Maschine wird gegen die Kreativität des Menschen ausgespielt, um eine Wieder-Verzauberung des Kunstwerks als unergründlich zu gewährleisten. Dabei ist die Maschine dort romantisch, wo sie gemäß der unendlichen Deutbarkeit und der Unergründbarkeit als KI zum Emblem dieses (Selbst-)Verständnisses wird.

Joachim Harst widmet sich in seinem Beitrag der postdigitalen Literatur Gregor Weichbrodts. In seinem Aufsatz »*What the Heck is a Book?*« *Code and Codex in Postdigital Literature* nimmt er Weichbrodts Text *I don't know* von 2014 zum Ausgangspunkt, die poetischen Verfahren algorithmischer Textproduktion zu untersuchen. Im Rückgriff auf den Begriff des »Postdigitalen« verbindet Harst das Algorithmische der Textproduktion mit dem Verhältnis von Code, Codex und Gesetz. Weichbrodts Text erscheint so als enzyklopädisch, insofern es Wissen und Nicht-Wissen ausstellt, das nur durch die algorithmische Verarbeitung der Wikipedia-

Datenbank erzeugt wird. Damit einher gehen eine subjektlose Autorschaft und eine Perspektive auf die Erhabenheit des Digitalen, die das Konzept ›Buch‹ auf die Probe stellt.

In ihrem Beitrag *Transitorische Literatur. Ebenen digitaler Schrift im Blog Ze zurrealism itself* begegnen Lore Knapp und Claus-Michael Schlesinger dem Problem des Transitorischen von Literatur auf Weblogs. Damit ist die Veränderbarkeit des einmal Geschriebenen gemeint, die das Schreiben im Internet erst ermöglicht. Sie gehen der Frage nach, wie die Textgenese eines fluiden digitalen Texts erfasst, beschrieben und analysiert werden kann. Es werden mehrere Ebenen digitaler Schriftlichkeit deutlich, die keineswegs einfach und vollständig erfasst werden können. Zwar lassen sich mithilfe des Internet Archives verschiedene Textstufen und -versionen rekonstruieren, aber ein ganzer Text bleibt eine Utopie. Vielmehr werden Praktiken des Schreibens im Internet sichtbar, die einen performativen Charakter besitzen.

Christian Schulz zeigt in seinem Beitrag *Vernakulärer Code oder die Geister, die der Algorithmus rief – digitale Schriftlichkeit im Kontext sozialer Medienplattformen*, dass sich digitale Schriftlichkeit nicht mehr nur im Backend finden lässt. Gerade beim Produzieren von Text auf sozialen Medienplattformen wird noch ein anderer Konnex deutlich: nämlich die Abhängigkeit von maschinellem Lernen, Algorithmen und Nutzer:innenverhalten. Schulz beschreibt das Interface, mit dem Nutzer:innen Text auf sozialen Netzwerken eingeben, als Schwelle gleich dreier Relationen, der Mensch-Maschine-, Maschine-Maschine- und einer Mensch-Mensch-Interaktion. Sichtbar werden diese Relationen und Interaktionen anhand von *Captions*. Unsichtbar dagegen sind diese dann, wenn tatsächlich Sichtbarkeit zum Thema wird und unter Umständen Postings und Kommentare nicht mehr im Feed erscheinen. Während *Captions* die Algorithmen ›füttern‹ und für Sichtbarkeit sorgen sollen, verweist das Nicht-Erscheinen von Posts im Feed auf die Interaktion von Algorithmen mit Nutzer:innen. Das Soziale ist von der Technik und damit von der digitalen Schriftlichkeit und *vice versa* abhängig.

Stefan Höltgen führt in seinem Beitrag *Humanities of the Digital. Philologische Perspektiven auf Source Codes als Beitrag einer computerarchäo-*

logischen Knowledge Preservation vor, wie eine Philologie der digitalen Schriftlichkeit aussehen kann. Programme und Programmiersprachen sind für Höltgen »Elaborate einer Textkultur«, die als kulturelle Artefakte erst philologisch als solche erfasst werden können. Anhand der Programmiersprache BASIC führt Höltgen vor, wie diese quellenhistorisch erfasst sowie sprach-, literatur- und bibliothekswissenschaftlich interpretiert werden kann. Erkennbar wird eine Programmierkultur, die mitnichten nur digital ist. Codes auf Papier, Notizen und Zeitschriften gehören genauso zum Aussagensystem BASIC wie die dazugehörigen Rechner, mit denen programmiert wurde und wird.

Als Begründer der *Critical Code Studies* führt Mark C. Marino in seinem Beitrag »*I hope you can read this*«. *Uncovering messages with Critical Code Studies* vor, wie eine solche Lektüre von Code eigentlich funktionieren kann. Er untersucht den Code der interaktiven digitalen Literatur *The Gay Science* von Capricorn van Knapp. Der Code Inform 7, der sich durch seine Nähe zur natürlichen Sprache Englisch auszeichnet, ist nicht nur der Hintergrund für die »interactive fiction«. Damit ein:e Leser:in das Spiel gewinnt, muss er/sie den Code lesen. Und weil Computercode eigentlich mehr für den Menschen als für den Computer geschrieben ist, wie Marino aufweist, zeigt sein Gegenstand, inwieweit der Code nicht nur Narrative, sondern auch Interaktionen bedingt. Dabei reicht es nicht, den Code wörtlich zu nehmen. Vielmehr bedarf es der literaturwissenschaftlichen Lese- und Interpretationstechniken, um den Code nicht nur in seiner Funktion für den Computer zu verstehen, sondern ihn auch in seinem sozio-kulturellen Kontext zu entschlüsseln. Daher plädiert Marino für einen interdisziplinären Ansatz, mit Informatiker:innen zusammenzuarbeiten und Code gemeinsam zu interpretieren. Denn jeder Code hat mehr an sich, als nur eine Maschine zum Ausführen eines Befehls zu bringen.

Der Band geht auf einen Online-Workshop zurück, der am 09.12. und 16.12.2022 an der Heinrich-Heine-Universität Düsseldorf stattfand. Die

Herausgeber danken der Universitäts- und Landesbibliothek Düsseldorf für die großzügige Förderung der Open-Access-Publikation. Wir danken ferner auch der Anton-Betz-Stiftung der Rheinischen Post e.V. für die Förderung der Drucklegung dieses Bandes.

Von ›bits‹ zu ›words‹

John von Neumanns *linguistic turn* als Ursprung digitaler Schriftlichkeit

Gabriele Gramelsberger

1. Einleitung

Digitale Schriftlichkeit als Programmieren, Prozessieren und Codieren von Schrift setzt bereits Schrift in zweifacher Weise voraus. Zum einen als materialisierte Schrift der Mechanisierung und Elektrifizierung der Sprache, die als logisch-elektronische Grundlage das Digitale erst ermöglicht; zum anderen als Maschinencode. Es ist die Idee des Maschinencodes, die John von Neumann als *linguistic turn* in die Instruierung elektrischer Computer Mitte der 1940er Jahre einführte und die das gesamte Potential des Phänomens der digitalen Schriftlichkeit überhaupt erst ermöglicht. Aus diesem *linguistic turn*, der von John von Neumann später kurzerhand »coding« genannt wurde, entwickelten sich ab Mitte der 1950er Jahre über erste Codierhilfen wie Assembler oder Compiler die höheren Programmiersprachen,¹ die nach über achttausend Jahren Schriftentwicklung einen neuen Schrifttyp darstellen.² Alle Phänomene des Digitalen – von der Automatisierung und Datafizierung bis hin zu

1 Herman H. Goldstine/John von Neumann: Planning and Coding Problems for an Electronic Computing Instrument, 1947, Part II, Vol 1. In: John von Neumann (Hg.): *Collected Works*, Bd. 5: *Design of Computers, Theory of Automata and Numerical Analysis*. Oxford: Pergamon Press 1963, 80–151, 103.

2 Achttausend Jahre Schriftentwicklung unter Berücksichtigung der in China gefundenen Jiahu-Zeichen (6.600 v. Chr.) und der Donauschrift (5.500 v. Chr.).

maschinellen Lernverfahren und Künstlicher Intelligenz – haben ihren Ursprung in diesem *linguistic turn*.

Die Bezeichnung *linguistic turn* im Zusammenhang mit John von Neumanns Einführung von Code und Coding ist eine retrospektive Zuschreibung der Computerhistoriker Mark Priestley und Thomas Haigh. In einem Beitrag über frühe US-amerikanische Computer stellten sie fest: »Computer programming took a linguistic turn, largely due to the fact of John von Neumann during the famous ›First Draft of a Report on the EDVAC‹ came up with the first thing that we would recognize as an order code, a machine programming linguistic coded instruction.«³ EDVAC [Electronic Discrete Variable Automatic Computer], dessen Design von Neumann 1945 beschrieben hatte, wurde zum Vorbild aller modernen Digitalcomputer. Mit EDVAC wurden Programme erstmals als Daten behandelt, binär codiert und im internen Speicher verarbeitet. Dieses Design wird bis heute von-Neumann-Architektur genannt. Den Bericht über das Design von EDVAC leitete von Neumann wie folgt ein:

»The considerations which follow deal with the structure of a very high speed automatic digital computing system, and in particular with its logical control. [...] These instructions must be given in some form which the device can sense: Punched into a system of punch-cards or on teletype tape, [...]. All these procedures require the use of some code, to express the logical and the algebraical definition of the problem under consideration, as well as the necessary numerical material.«⁴

Damit war die Bühne von Hardware und Software eröffnet, auf welcher sich von da an – theatral formuliert – die Dramen des Digitalen abspielen sollten. Vor dem Hintergrund der Wirkmächtigkeit dieses *linguistic*

3 Mark Priestley/Thomas Haigh: Working on ENIAC: The Lost Labors of the Information Age. In: *MITH Digital Dialogues on opentranscripts.org*, 02.18.2016. <http://opentranscripts.org/sources/mith-digital-dialogues/> (zuletzt abgerufen am 09.06.2023).

4 John von Neumann: *First Draft of a Report on the EDVAC*. University of Pennsylvania: Moore School of Electrical Engineering 1945, 1.

turns ist die Analyse der verschiedenen Arten des Schriftgebrauchs für ein kritisches Verständnis der digitalen Schriftlichkeit von Interesse.

2. Mechanisierung der Sprache

Der Digitalcomputer, das wird gern vergessen, ist der materialisierte Endpunkt einer jahrhundertealten Entwicklung der Mechanisierung und Elektrifizierung der Sprache.⁵ Zu Beginn dieser Entwicklung stand das philosophische Programm der Operationalisierung des Geistes. Rationalisten wie René Descartes und Gottfried Wilhelm Leibniz als auch Empiristen wie John Locke und David Hume widmeten sich der Analyse und Operationalisierung der Erkenntnisfunktionen des Subjekts. Von Descartes' analytischem Problemlösungsverfahren über Lockes Definition von Wissen als »die Auffassung der Verbindung und Übereinstimmung oder der Nichtübereinstimmung und des Widerstreits unserer einzelnen Vorstellungen [...]. Darin allein besteht es«⁶ bis hin zu Leibniz' Ersetzung von Ähnlichkeit durch Äquivalenz (*salva veritate*) formierte sich die konzeptuelle Grundlage des Digitalen bereits in der Neuzeit. Insbesondere Leibniz hatte nicht nur ein Faible für Logik und Kalküle und erfand das binäre Rechnen, sondern er war sich schon 1693 über die Tragweite einer an Maschinen delegierten Sprache in Form von Kalkülen bewusst. Sie wäre zukünftig eine »Ergänzung der sinnlichen Anschauung und gleichsam ihre Vollendung« und würde auch »in den Beschreibungen der Mechanismen der Natur« äußerst nützlich sein.⁷

Was mit den Analysen der Rationalisten und Empiristen entstand, war ein formales Verständnis von Verstandesfunktionen, das von allem Inhaltlichen abstrahierend die Verstandesfunktionen als Operationen

5 Gabriele Gramelsberger: *Philosophie des Digitalen. Zur Einführung*. Hamburg: Junius 2023.

6 John Locke: *Versuch über den menschlichen Verstand*. Berlin: Heimann 1872, 4. Buch, Kap. 1, §2.

7 Gottfried Wilhelm Leibniz: De analysi situs/Kalkül der Lage. In: Ders.: *Philosophische Werke*, 1. Bd. Hamburg: Meiner 1996, 69–76, 76.

fasste. Bei aller Differenz gelangen beide philosophische Strömungen am Ende zu sehr ähnlichen Operationsmodellen des Denkens. So waren sie sich einig, dass Verstandesfunktionen in sehr einfachen Operationen des Vergleichens, Feststellens von Identität oder Nicht-Identität, des Auseinanderdividierens (Analyse) und des Zusammensetzens (Synthese) bestanden. Das Feststellen von Identität oder Nicht-Identität (Negation) sowie das Zusammensetzen in Form der Konjunktion und Disjunktion konstituieren das Spektrum der aussagenlogischen Verknüpfungen, die heute die primitiv-rekursive und entscheidungsdefinite Grundlage des Digitalen auf Basis der Booleschen Algebra (AND, OR, NOT) bilden. Hinzukommt, dass mit Leibniz' Prinzip *salva veritate* sich eine Aussage durch eine andere Aussage formal ersetzen lässt, wenn dadurch der Wahrheitswert der Aussage oder des Satzes nicht verändert wird.

Doch auch wenn sich die Verstandesfunktionen als Operationen bereits im 16. und 17. Jahrhundert abstrahieren ließen, so fehlte es an einem adäquaten Ausdrucksmittel. Denn seit der Antike nutzte die Philosophie die aristotelische Logik (Syllogistik), die als eine Sprachlogik auf semantische Inhalte konzentriert war. Erst George Boole gelang es in seinem epochalen Werk *The Mathematical Analysis of Logic* von 1847, ein adäquates Ausdrucksmittel zu entwickeln. Boole hatte nichts geringeres als die Formalisierung der Logik selbst zum Ziel. Dazu analysierte er die Struktur der Urteilsformen der aristotelischen Syllogistik, um zu seinem »Calculus of Logic« zu gelangen; »a method resting upon the employment of Symbols, whose laws of combination are known and general, and whose results admit of a consistent interpretation.«⁸ Die Gleichsetzung von Logik und Zeichengebrauch, die heute symbolische Logik genannt wird, erforderte aber die Einführung einer neuen Operation: eines formalen Mechanismus der Klassifikation und der Auswahl von Individuen. Klassenbildung respektive Selektion, so Boole, sollten nicht mehr länger ein Geschäft der aristotelischen Metaphysik, sondern der Mathematik sein. Erst dieser Selektionsmechanismus ermöglichte es Boole, die Struktur

8 George Boole: *The mathematical analysis of logic, being an essay towards a calculus of deductive reasoning*. Cambridge: Macmillan, Barclay, & Macmillan u.a. 1847, 4.

der aristotelischen Logik zu formalisieren sowie die logischen Urteilsformen als mathematische Gleichungen darzustellen.⁹ Aus den aristotelischen Urteilsformen der allgemein bejahenden oder verneinenden Urteile sowie der partikular bejahenden oder verneinenden Urteile wurden Formeln: Aus »Alle X sind Y« wurde bei Boole » $xy = x$ «, aus »Kein X ist Y« wurde » $xy = 0$ «, aus »Einige X sind Y« wurde » $v = xy$ « und aus »Einige X sind nicht Y« wurde » $v = x(1-y)$ «. Mit diesen Formeln ließ sich nun rechnen. Auf dieser Basis rekonstruierte Boole die aristotelische Logik als formale Aussagenlogik, die später axiomatisiert zur booleschen Algebra wurde, die bis heute die logische Grundlage der Schaltungen von Digitalcomputern bildet. Die boolesche Algebra ist damit der zentrale Markstein der Mechanisierung der Sprache. Allerdings, und darauf hatte etwas später Charles S. Peirce aufmerksam gemacht, formalisierte Booles Logikkalkül nur einen Bruchteil logischer Schlussformen der aristotelischen Syllogistik. Daher erweiterte Peirce später Booles Kalkül zu einem Relationskalkül und entwarf damit eine erste Prädikatenlogik.¹⁰

3. Elektrifizierung der Sprache

Doch Mechanisierung der Sprache alleine genügt nicht, um diese an Maschinen wie den Digitalcomputer zu delegieren. Hier kommt ein weiterer Schriftgebrauch ins Spiel, der die Elektrifizierung der Sprache ermöglichte und ein Jahrhundert nach Boole von Claude Shannon ins Spiel gebracht wurde. Im Kontext der Telefonie entwickelte sich zu Beginn des 20. Jahrhunderts die Theorie der gepulsten Signalübertragung von

9 Was die Logik hierfür prädestinierte, war aus Booles' Perspektive Folgendes: »That which renders Logic possible, is the existence in our minds of general notions, – our ability to conceive of a class, and to designate its individual members by a common name. The theory of Logic is thus intimately connected with that of Language.« Boole: *Mathematical Analysis of Logic*, 4–5.

10 Charles S. Peirce: Description of a Notation for the Logic of Relatives, Resulting from an Amplification of the Conceptions of Boole's Calculus of Logic. In: *Memoirs of the American Academy of Sciences* 9 (1870), 317–378.

Sprache. Die Frage, die es dabei zu lösen galt, war, welche diskrete Übertragungsrate die effizienteste für ein analoges Sprachsignal wäre, ohne Informationen der Sprachübertragung zu verlieren. Die Bell Lab Forscher Harry Nyquist und Ralph Hartley entwickelten eine Theorie der Abtastung und Quantisierung (Quantisierungsstufen) kontinuierlicher Signale für die elektrische Kommunikation. Um den Übergang vom physikalischen Phänomen elektromagnetischer Trägerwellen zum Symbol zu fassen, bedarf es des Wechsels von Sinuswellen zu Logarithmen sowie der Zuordnung von reellen Zahlen (*digits*) zu den einzelnen Quantisierungsstufen. Auf diese Weise lässt sich ein kontinuierliches Signal in ein Digitalsignal transformieren, welches jeder Quantisierungsstufe einen eindeutigen (reellen) Zahlenwert zuordnet.

Im Sinne einer effizienten Übertragung von Informationen führte Shannon 1948 einen statistisch-stochastischen Ansatz in die elektrische Informationsübertragung ein, indem er die Informationsentropie (H) als Maßeinheit für den Informationsgehalt einer Symbolfolge definierte. Dabei betrachtete Shannon den Informationsgehalt einer Symbolfolge rein statistisch als Häufigkeitsvorkommnis von Sprachzeichen und Zeichenkombinationen. In natürlichen Sprachen wie dem Englischen ist folgendes offensichtlich: »The letter E occurs more frequently than Q, the sequence TH more frequently than XP etc.«¹¹ Der entscheidende Kunstgriff, den Shannon anwandte, war aber folgender: Er stellte Informationen mit Markov-Ketten dar, um die Wahrscheinlichkeit des Vorkommens eines Symbols in einer Symbolreihe sowie die Wahrscheinlichkeit der Abhängigkeit des Vorkommens eines Symbols von den vorhergehenden vorherzusagen. Je komplexer das Vorhersagemodell (Markov-Kette), desto mehr nähert sich die Vorhersage den tatsächlichen Symbolvorkommnissen an. Auf dieser Basis kann Shannon nun die Entropie einer Symbolfolge (Nachricht) bestimmen, um so die notwendigen Übertragungskapazitäten in Form von Bits pro Sekunde (*binary digit*) zu berechnen. Eine hohe Entropie bedeutet eine hohe Redundanz, also statistische

11 Claude Shannon: A Mathematical Theory of Communication. In: *Bell System Technical Journal* 27 (1948), 379–423 und 623–656, 385.

Regelmäßigkeit, in einer Symbolfolge. Ein Werk wie James Joyces *Finnegans Wake* hat eine niedrige Entropie im Unterschied zu *Basic English*, das nur etwa 850 Wörter umfasst. Je redundanter eine Nachricht ist, desto geringer muss die Kapazität für eine verlustfreie Datenübertragung sein und desto weniger Bits werden für die Codierung einer Nachricht benötigt. Eine Nachricht, bestehend aus zwei gleichwahrscheinlichen Zeichen, hat nach Shannon den Informationsgehalt $\log_2(\frac{1}{2})$ und benötigt 1 Bit zur Übertragung. Ob ein Bit (Binärzahl) mit den Ziffern 0 und 1 dargestellt wird, hängt von der gewählten Konvention des Alphabets ab.

Shannons Wechsel von der Analyse zur Vorhersage erforderte die Kenntnis über die typischen Wahrscheinlichkeiten des Vorkommens von Symbolen in einer Sprache. Andrej A. Markov hatte dies für die Buchstabensequenzen in der russischen Literatur Ende des 19. Jahrhunderts untersucht.¹² Friedrich W. Kaeding hatte die Häufigkeit von Buchstaben, Silben und Wörtern mit einem Team von über eintausend Freiwilligen für die deutsche Sprache analysiert und 1898 publiziert.¹³ Am Ende hatte Kaeding ein Korpus von über 10 Millionen Wörter und 60 Millionen Buchstaben zusammengetragen; ein Korpus, das in einem derartigen Umfang erst wieder ab den 1970er Jahren mit Hilfe von Computern generierbar wurde. Diese quantitativ-statistischen Analysen der Sprache waren nicht nur Vorläufer der heutigen Digital Humanities, sondern ermöglichten erst die Verbindung von elektrischer Energie und Symbol.

Beide Arten des (Schrift-)Sprachgebrauchs, der logische und der statistisch-stochastische, verbinden sich in Flip-Flop-Schaltungen

-
- 12 Andrej A. Markoff: *Wahrscheinlichkeitsrechnung*. Übers. v. Heinrich Liebmann. Leipzig: Teubner 1912; Philipp von Hilgers/Vladimir Velminski (Hg.): *Andrej A. Markov. Berechenbare Künste*. Zürich/Berlin: diaphanes 2007.
- 13 Friedrich W. Kaeding: *Häufigkeitswörterbuch der deutschen Sprache: Festgestellt durch einen Arbeitsausschuss der deutschen Stenographiesysteme*. Steglitz bei Berlin: Selbstverlag des Herausgebers/E.S. Mittler & Sohn 1898; Toni Bernhart: Von Aalschwanzspekulanten bis Abendrotlicht. Buchstäbliche Materialität und Pathos im Häufigkeitswörterbuch der deutschen Sprache von Friedrich Wilhelm Kaeding. In: Ralf Klausnitzer/Carlos Spoerhase/Dirk Werle (Hg.): *Ethos und Pathos der Geisteswissenschaften*. Berlin/Boston: De Gruyter 2015, 165–190.

(on/off) als Grundeinheiten der elektrifizierten Information, insofern, wie Shannon schrieb, »a relay or a flip-flop circuit can store one bit of information.«¹⁴ Flip-Flop-Schaltungen sind aus mehreren Logikgattern zusammengesetzte Schaltungen. Logikgatter wiederum sind Anordnungen von Schaltungen zur (binären) Realisierung boolescher Funktionen gemäß Shannons Schaltalgebra von 1938, die die logischen Operatoren AND, OR, XOR sowie deren Negationen repräsentieren.¹⁵ Aus diesen Schaltungen lässt sich die gesamte arithmetische und logische Funktionalität digitaler Computer wie auch ihre Datenprozessierungs- und -speicherkapazitäten konstruieren.

4. John von Neumanns *linguistic turn*

Digitale Schriftlichkeit gründet in diesen beiden, hoch technisierten Arten des (Schrift-)Sprachgebrauchs, die jedoch am Ende nur Signale oder Bits übriglassen und daher einer mnemotechnischen Erschließung bedürfen. Eine solche mnemotechnische Erschließung ist notwendig, um den Vorteil der freien Programmierbarkeit von Computern als allgemeinen Maschinen nutzen zu können. Ohne diese Erschließung ist die Eingabe eines Programms mühsam, denn für jede neue Aufgabenstellung mussten die zahlreichen Schaltungen der ersten elektrischen Computer einzeln per Hand verbunden werden. Beispielsweise mussten die 17.468 Elektronenröhren, 7.200 Dioden, 1.500 Relais, 70.000 Widerstände und 10.000 Kondensatoren der Rechen-, Zähl- und Speichereinheiten von ENIAC (Electronic Numerical Integrator and Computer), der Vorläufer von EDVAC, für jede neue Berechnung neu verkabelt werden. »Setting up the ENIAC meant plugging and unplugging a maze of cables and setting arrays of switches. In effect, the machine had to be rebuilt for each

14 Shannon: A Mathematical Theory of Communication, 379.

15 Claude Shannon: A Symbolic Analysis of Relay and Switching Circuits. In: *Transactions of the American Institute of Electrical Engineers* 57 (1938), 38–80.

new problem it was to solve.«¹⁶ Während das Verkabeln (Programmeingabe) per Hand Tage dauerte, konnte ENIAC, basierend auf Daten, die mit Lochkarten eingegeben wurden, komplexe Berechnungen in Stunden oder sogar Minuten lösen. Doch auch die Zerlegung von Berechnungen per Hand in Teilberechnungen und schaltungsgerechte Abläufe sowie die Übersetzung in einen Plan der Verkabelung, dauerten ihre Zeit, ebenso wie die Fehlersuche und -korrektur. ENIAC, so wird kolportiert, rechnete nur rund zwei Stunden pro Woche. Der Rest der Zeit musste in die Verkabelung wie auch Fehlersuche investiert werden.

Eben diese Situation fand John von Neumann vor, als er sich als Mathematiker mit dem Bau aber auch dem Instruieren der ersten elektronischen Computer in den USA beschäftigte. Von Neumann, ein ungarischer Mathematiker, der bei David Hilbert in Göttingen studiert und sich in Berlin habilitiert hatte, emigrierte 1929 in die USA. Er war während des Zweiten Weltkrieges in Los Alamos im Manhattan-Projekt für die Berechnung der Ausbreitung von Explosionswellen zuständig. Allerdings erwiesen sich die Ausbreitungsmodelle basierend auf partiellen Differentialgleichungen als zu komplex, als dass sie per Hand analytisch lösbar waren. Der Mathematiker Stanislaw Ulam beschrieb die Situation 1943 in Los Alamos wie folgt:

»The blackboard was filled with very complicated equations that you could encounter in other forms in other offices. [...] looking at these I felt that I should never be able to contribute even an epsilon to the solution of any of them. But during the following days, to my relief, I saw that the same equations remained on the blackboard. I noticed that one did not have to produce immediate solutions.«¹⁷

Vor diesem Hintergrund erkannte von Neumann nicht nur die Bedeutung freiprogrammierbarer Digitalrechner, sondern entwickelte Metho-

16 Paul E. Ceruzzi: *A History of Modern Computing*. Cambridge, MA: The MIT Press 1998, 21.

17 Stanislaw Ulam: Von Neumann: The Interaction of Mathematics and Computing. In: Nicholas Metropolis/Jack Howlett/Gian-Carlo Rotta (Hg.): *A History of Computing in the Twentieth Century*. New York: Academic Press 1980, 93–99, 95.

den zur Diskretisierung von Differentialgleichungsmodellen, um diese numerisch zu berechnen respektive die Berechnungen an Computer delegieren zu können. Heute nennen wir diese Methode numerische Simulation partieller Differentialgleichungsmodelle, beispielsweise von Wetter- oder Klimamodellen – oder kurz: Computersimulation.¹⁸

Die Mühsamkeit der Verkabelung von ENIAC zur Berechnung kurzer Programme einerseits sowie die Notwendigkeit andererseits, immer umfangreichere Berechnungen für Differentialgleichungsmodelle durchzuführen, waren der Grund, warum von Neumann über »a very high speed automatic digital computing system, and [...] its logical control« nachdachte.¹⁹ Er entwarf nicht nur die bis heute verwendete Architektur von Digitalcomputern, die Programme als Daten behandelte und über Lochkarten einlesbar oder sogar intern speicherbar machte, sondern vollzog für seine neue Maschine einen *linguistic turn*.²⁰ Konkret meint dies die Verwendung von Codeworten für die logische Kontrolle der Maschine. »It is therefore our immediate task to provide a list of the orders which control the device, i.e. to describe the code used in the device, and to define the mathematical and logical meaning and the operational significance of its code words.«²¹ Von Neumann gibt am Ende seines Designs von EDVAC eine Liste von Codeworten und deren Bedeutung. Beispielsweise meint das Codewort »i₁=1« in seiner Kurzform »whup« den Befehl »to carry out the operation w in CA and to

18 Der Begriff Computersimulationen reicht über die numerische Simulation partieller Differentialgleichungen (deterministische Simulation) hinaus und bezeichnet weitere Simulationsformen wie stochastische Simulationen (z.B. Monte-Carlo-Simulation). Gabriele Gramelsberger (Hg.): *From Science to Computational Sciences. Studies in the History of Computing and its Influence on Today's Sciences*. Zürich/Berlin: diaphanes 2011.

19 von Neumann: *Report on the EDVAC*, 1. Von Neumann konzipierte Computer als Maschinen mit Rechenwerk (arithmetisch-logische Einheit), Steuerwerk, Bus-einheit (Daten- und Energieübertragung zwischen den Komponenten), Speicherwerk sowie Eingabe-/Ausgabewerk.

20 Thomas Haigh/Mark Priestley/Crispin Rope: *ENIAC in Action: Making and Remaking the Modern Computer*. The MIT Press: Cambridge, MA 2016.

21 von Neumann: *Report on the EDVAC*, 85. Unterstreichungen im Originaltext.

dispose of the result. [...] h means that the result is to be held in O_{ca} . up means, that the result is to be transferred into the minor cycle ρ in the minor circle u.«²²

Es ist die symbolische Ersetzung X (*words*) für die Maschinenoperation Y, die den *linguistic turn* markiert. Was heute selbstverständlich anmutet, war 1945 neu und revolutionierte die Computerentwicklung. Denn diese symbolische Ersetzung (Maschinencode für Maschinenoperationen) vereinfachte nicht nur die Eingabe eines Programms in einen Computer, sondern stellte eine erste Form der Automatisierung dar und zwar dessen, was bis dahin per Hand zur Verkabelung einer Maschine ausgeführt werden musste. 1951 ging Maurice Wilkes, der an der Entwicklung des Manchester Computers beteiligt war, einen entscheidenden Schritt weiter. Er schlug vor, basale Schalt-, Rechen- und Steuerabläufe unterhalb der Ausführungsebene von Maschinencodes festzulegen, und nannte diese unveränderbare Form des Codes »micro-operations« (Microcode): »Each true machine operation is thus made up of a sequence or ›micro-program‹ of micro-operations. [...] Only 40 micro-orders are required to perform all these operations [add, subtract, multiply, transfer etc.].«²³ Microcode wird in der Regel fest in einen Computer respektive Prozessor integriert. Maschinencodes adressieren dann die Microcodes, die lediglich Abfolgen von Nullen und Einsen sind.

Die mnemotechnische Revolution des Maschinencodes initiierte eine Kaskade an Folgephänomenen, in deren Verlauf sich die symbolische Ersetzung immer mehr an den menschlichen Sprachgebrauch annäherte. Compiler waren erste Programme, die Programmcode in Maschinencode übersetzten, die dann in einem Computer wiederum via Microcodes in Maschinenoperationen transferiert wurden. Allerdings wurden diese ersten Programmierhilfen nicht von allen begeistert aufgenommen. »At that time [1954], most programmers wrote symbolic machine instructions exclusively [...] they firmly believed that any mechanical coding method would fail to apply the versatile ingenuity

22 von Neumann: *Report on the EDVAC*, 100.

23 Maurice V. Wilkes: *The Best Way to Design an Automated Calculating Machine*. In: *Manchester University Computer Inaugural Conference* (1951), 182–184, 183–184.

which each programmer felt he possessed and constantly needed in his work.«²⁴ Diese ersten Programmierhilfen entwickelten sich Mitte der 1950er Jahre zu Programmiersprachen, die einerseits weitere Vereinfachungen des Programmierens zum Ziel hatten (Automatisierung), andererseits aus der Kombination einfacher Operationen zunehmend komplexere Programmierbefehle ermöglichten. Dadurch konnte das Möglichkeitspotential des frei programmierbaren Digitalcomputers immer besser ausgeschöpft werden. John Backus beispielsweise, der die höhere Programmiersprache FORTRAN (Formula Translation) entwickelte und 1954 vorstellte, stellte sich folgende Frage:

»What could be done now to ease the programmer's job? Once asked, the answer to this question had to be: Let him use mathematical notations. But behind that answer [...] there was the really new and hard question: Can a machine translate a sufficiently rich mathematical language into a sufficiently economical machine program to make the whole affair feasible?«²⁵

Backus' letzte Frage ist nicht trivial, denn es muss sichergestellt sein, dass der Computer tatsächlich die programmierte Berechnung ausführt. Solange die Programmstrukturen dies direkt in Maschinencode übertragen, kann er oder sie dies selbst überprüfen. Wird die Übersetzung jedoch durch eine Programmiersprache via einen Compiler oder einen Interpreter automatisiert, muss man sich auf diese Programmierhilfen verlassen. Die Entwicklung von Programmierhilfen und Programmiersprachen stellte die erste Welle der Automatisierung durch das Digitale dar. Viele weitere sollten folgen bis hin zum aktuellen Hype der Künstlichen Intelligenz.

24 John Backus/William P. Heising: FORTRAN. In: *IEEE Transactions on Electronic Computing* 13 (1964), 382–385, 382.

25 John Backus: Programming in America in the 1950s. In: Nicholas Metropolis/Jack Howlett/Gian-Carlo Rotta (Hg.): *A History of Computing in the Twentieth Century*. New York: Academic Press 1980, 125–135, 131.

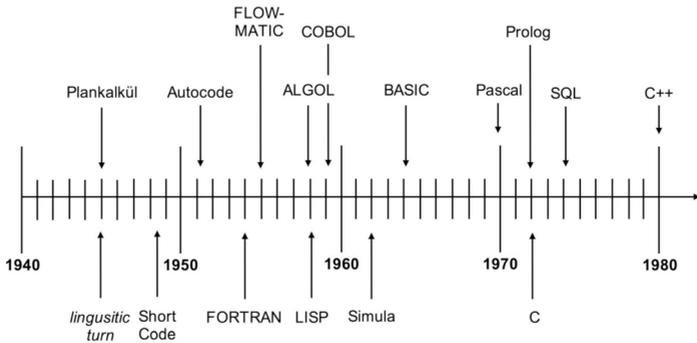
5. Protodigitale Schriftlichkeit

Mit dem Übergang von Maschinenoperationen und Microcode zu Maschinencode und später zur Automatisierung durch die zunehmend komplexeren Programmcodes der Compiler und höheren Programmiersprachen setzte sich nach achttausend Jahren Schriftentwicklung ein neuer Typ von Schriftgebrauch durch: autooperative Schriften, die einem Zeichengebrauch folgen, »den man den deklarativen Zeichengebrauch nennen könnte, bei dem ein Autor eine Aufgabe anschreibt (deklariert), die dann von einer Maschine exekutiert wird.«²⁶ Digitale Schriftlichkeit basiert auf diesem deklarativen Zeichengebrauch, der wiederum in dem skizzierten Programm der Mechanisierung und Elektrifizierung der Sprache gründet. Doch digitale Schriftlichkeit verlässt sich auch auf den skizzierten *linguistic turn*, insbesondere auf dessen erste Errungenschaft: Codeworte, Microcodes und schließlich Maschinencodes, die sich als Formen einer protodigitalen Schriftlichkeit charakterisieren lassen. Das Adjektiv ›protodigital‹ bezieht sich darauf, dass Codeworte, Microcodes und Maschinencodes eine notwendige Vorform der digitalen Schriftlichkeit sind, gleichwohl es sich um digitale Anweisungen handelt (Folgen von Bytes, die sowohl Daten als auch Befehle repräsentieren). Deutlich wird dies auch durch das Adjektiv ›höhere‹, das das Protodigitale insofern in sich trägt, als Programmiersprachen nicht unmittelbar von Computerprozessoren verstanden und ausgeführt werden können, sondern interpretiert oder kompiliert werden müssen. Diese Differenz im Schriftgebrauch hat zur Folge, dass erst höhere Programmiersprachen die gesamte Vielfalt an Möglichkeiten der digitalen Operativität eröffnen, aber zum Preis der Spezialisierung auf Teilbereiche wie die mathematische Operativität (FORTRAN), die logische Operativität (Prolog), die Operativität der Datenverwaltung (SQL) und viele weitere Formen der Operativität. Die maschinischen Grundoperationen respektive Microcodes fungieren

26 Gernot Grube: Autooperative Schrift. In: Ders./Werner Kogge/Sybille Krämer (Hg.): *Kulturtechnik Schrift. Die Graphé zwischen Bild und Maschine*. München: Fink 2005, 81–114, 82.

dabei wie ein Baukasten, mit dem sich durch geschickte Kombinationen zunehmend komplexere, aber eben auch spezialisiertere Programmierbefehle kombinieren lassen. Wenn also von digitaler Schriftlichkeit die Rede ist, sind einerseits diese maschinisch zugerichteten Schriftvorsetzungen, aber andererseits auch die Ambiguität einer zunehmenden Erweiterung digitaler Operativität bei gleichzeitiger Spezialisierung mitzudenken.

Abb. 1: Entwicklung von frühen Programmierhilfen und -sprachen²⁷



Doch bereits in den 1960er und 1970er Jahren zeigte sich, dass für von Neumanns wegweisende Konzepte, sowohl die des Computers wie

27 Parallel zu John von Neumann hatte sich der Berliner Konrad Zuse zwischen 1942 und 1945 Notizen zu einer höheren Programmiersprache (Plankalkül) für seinen geplanten Computer Z3 gemacht, konnte diesen jedoch aufgrund des Krieges nicht verwirklichen. Der Plankalkül wurde erst 1972 vollständig veröffentlicht und ist Anfang der 2000er Jahre erstmals rekonstruiert und implementiert worden. Konrad Zuse: Über den allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben. In: *Archiv der Mathematik* 1 (1948/49), 441–449; Friedrich L. Bauer/Hans Wössner: The »Plankalkül« of Konrad Zuse: A Forerunner of Today's Programming Languages. In: *Communications of the ACM* 15 (1972), 678–685.

seiner mnemotechnischen Revolution (*linguistic turn*), ein hoher Preis zu zahlen war. Backus hat dies 1978 treffend formuliert:

»In order to understand the problems of conventional programming languages, we must first examine their intellectual parent, the von Neumann computer. [...] In its simplest form a von Neumann computer has a central processing unit (or CPU), a store, and a connecting tube that can transmit a single word between the CPU and the store (and send an address to the store). I propose to call this tube the *von Neumann bottleneck*. The task of a program is to change the contents of the store in some major way; when one considers that this task must be accomplished entirely by pumping single words back and forth through the von Neumann bottleneck, the reason for its name becomes clear. Ironically, a large part of the traffic in the bottleneck is not useful data but merely names of data, as well as operations and data used only to compute such names.«²⁸

Es ist klar, dass dieser von-Neumann-Flaschenhals die Programmausführung erheblich verlangsamte. Dies wurde und wird zwar durch zunehmend kleinere und schnellere Mikroprozessoren kaschiert. Doch das ist nur ein Aspekt. »More importantly, it is an intellectual bottleneck that has kept us tied to a-word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand.«²⁹ Das Problem wird bis heute durch die Entkopplung von Speicher und Rechenwerken über Hierarchien an Zwischenspeicher praktisch gelöst. Maurice V. Wilkes, der das Konzept der Zwischenspeicher entwickelte, motivierte seine »slave memories« wie folgt:

»The use is discussed of a fast core memory of, say, 32000 words as a slave to a slower core memory of, say, one million words in such a way that in practical cases the effective access time is nearer that of the fast

28 John Backus: Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs. In: *Communications of the ACM* 21 (1978), 613–641, 615.

29 Backus: Can Programming Be Liberated, 615.

memory than that of the slow memory. [...] By a slave memory I mean one which automatically accumulates to itself words that come from a slower main memory, and keeps them available for subsequent use without it being necessary for the penalty of the main memory access to be incurred again.«³⁰

Bis heute arbeitet sich die Entwicklung von Computern wie auch Programmiersprachen am von-Neumann-Flaschenhals in der einen oder anderen Weise ab, auch wenn Prozessoren unglaublich klein (2nm), schnell (ExaFlop/s) und komplex (*system on a chip*) geworden sind und die Datenvolumen unfassbare Ausmaße (Zettabytes) angenommen haben. Doch wie umfangreich die digitale Schriftlichkeit, die uns als sichtbares Phänomen auf der Oberfläche digitaler Geräte im Alltag begleitet, auch immer ist, an der konzeptuellen Grundlage ihrer Programmierung, Prozessierung und Codierung hat sich bis heute nicht viel geändert, solange von-Neumann-Computer verwendet werden. Allerdings macht sich mittlerweile ein Effekt bemerkbar, der im Paradox des *linguistic turn* seinen Ursprung hat. Dieses Paradox resultiert aus dem Umstand, dass der *linguistic turn* Maschinenoperationen mnemotechnisch erschließbar macht und so die Nähe zur Sprache wieder herstellt, allerdings zum Preis eines sehr aufwendigen »a-word-at-a-time thinking«, dass zur Produktion von Unmengen an Daten führt und immer mehr Speicher und Computerleistung erfordert.³¹ Diese mnemotechnisch Revolution ermöglichte jedoch erst die digitale Revolution gerade wegen des anthropomorphen (Schrift-)Zugangs zum Digitalen.

30 Maurice V. Wilkes: Slave Memories and Dynamic Storage Allocation. In: *IEEE Transactions on Electronic Computers EC-14* (1965), 270–271, 270. Das Konzept der Zwischenspeicher wie auch der Trennung von Programm- und Datenzwischenspeichern hat neben dem von-Neumann-Computer (SISD-Computer, *Single Instruction, Single Data*) zu neuen Computerarchitekturen wie Vektorrechnern (SIMD-Computer, *Single Instruction, Multiple Data*) oder Parallelrechnern (MIMD-Computer, *Multiple Instruction, Multiple Data*) geführt. Michael J. Flynn: Some Computer Organizations and Their Effectiveness. In: *IEEE Transactions on Computers C-21* (1972), 948–960.

31 Backus: Can Programming Be Liberated, 615.

Doch dieser anthropomorphe Zugang droht zunehmend verloren zu gehen. Denn die Unmengen an ikonoklastischen Daten wie auch die enormen Rechengeschwindigkeiten von Billionen von Operationen pro Sekunde unterlaufen schlichtweg die menschlichen Wahrnehmungsfähigkeiten. Hinzu kommt der zunehmende Einsatz von KI (Künstlicher Intelligenz) zur automatisierten Kontrolle und Steuerung des Digitalen. Zusammengenommen erleben wir heute einen Effekt der Deanthropomorphisierung des Digitalen, der uns immer weniger direkten Zugang erlaubt.³² Im Zuge dieser Deanthropomorphisierung geht auch der mnemotechnische Zugang durch von Neumanns *linguistic turn* verloren. Nicht nur weil Programmierung zunehmend an KI delegiert wird, sondern weil die Milliarden smarterer Objekte sich effizienter in binär-codierten Microcodes oder elektrischen Signalen als in symbolischer Maschinensprache unterhalten. Der Umweg über mnemotechnische Hilfen von Codeworten und den *linguistic turn* ist aufgrund der unglaublichen Automatisierungskomplexität und -tiefe des Digitalen und dessen Autooperativität nicht länger notwendig. Sie ist allenfalls eine nostalgische Reminiszenz an den Menschen.

32 Eine ausführliche Diskussion dieser Entwicklung findet sich in Gramelsberger: *Philosophie des Digitalen*.

(Elektrographisches) Schreiben als Praxis in Bild, Schrift und Material

Tilo Reifenstein

Wenn Sie, als Leser:in dieses Textes in Analogform, einmal kurz mit Ihrem Telefon von dieser Seite ein Bild machen, bieten Ihnen Google Lens und Apples Live Text sogleich die Möglichkeit, das Schriftbild von seiner Bildlichkeit zu trennen und in wandelbaren Fließtext umzuformen. Begegneten Sie dem Text in digital Form, ist Vergleichbares über eine Bildschirmaufnahme möglich, obwohl Sie, je nach zugrundeliegendem Dateiformat und -eigenschaften, eventuell auch schon direkten Zugriff auf kopierbaren Rohtext haben.

Abhängig vom originalen Schriftbild, der Fotoqualität und spezifischen Umsetzung ordnen Texterkennungsprogramme graphisch-materiellen Elementen bestimmte Werte (Symbole, Buchstaben, Zahlen) üblicher elektronischer Textcodierung (Unicode oder auch Untergruppen wie ASCII) zu und können diese entsprechend dem Ausgabeformat mit Layoutinformationen speichern. So wird das Rasterbild normalerweise auf Störsignale und etwaige Hintergrundfarben hin untersucht, um ›Zeichenrelevantes‹ von ›unrelevantem‹ zu trennen und visuellen Reichtum auf Pixelebene zu reduzieren und ins Schwarz-weiß zu übertragen. Was eben noch Bild war, wird nun auf Verkrümmungen und -zerrungen im erahnten Text überprüft. *Zoning* isoliert Überschriften, Absätze, Bilder, Tabellen und andere Formatierungselemente. Durch Segmentierungsprotokolle werden schließlich Absätze in Textzeilen, dann durch Weißstellen getrennte Worte und Buchstaben aufgebrochen. Bei *optical-character-recognition* (OCR) Programmen, die mit *matching* arbeiten, laufen einzelne Zeichenmustererkennungspro-

zesse ab, die Größe und Struktur untersuchen und mit Bibliotheken abgleichen. Anwendungen auf der Ebene der *feature recognition* brechen das vermutliche Zeichen weiter auf, um Schlaufen, Bögen und Strichüberschneidungen als Komponenten spezifischer Werte zu überführen. Der so transkribierte Rohtext wird in der Regel durch Verwendung von Lexika sowie linguistische und stochastische Verfahren nachbearbeitet.

Einer materiellen graphischen Markierung auf einer analogen Buchseite wird derart entweder ein eindeutiger Zeichenwert eines bereits existierenden Modells zugeteilt oder nicht. Etwas wird erkannt oder dem Nichtschriftlichen zugeordnet. Die Möglichkeit der Fehlübertragung ist hier ebenso betonenswert, wie die generelle Unterwerfung von etwas Geschriebenem unter vorformulierte Schemata. Die unbequeme Umklammerung innerhalb der Überschrift oder etwaige Marginalien, Durch- und Unterstreichungen, Hervorhebungen, nicht-lineare Layouts, Eselsohren und so weiter, erschweren dabei eine solche Übertragung. Johanna Drucker fasst bezüglich der Texterkennung einer alphabetischen Quelle pointiert zusammen, was auch für andere Schreibformen gilt:

»whether the methods are statistical, heuristic, or based in neural networks and other protocols, they all conceive of the alphabet as a set of graphically discrete elements whose identity resides in their capacity to be differentiated one from another.«¹

Für Drucker erfolgt diese Beobachtung in der Coda ihres Buches zum Ursprung des Alphabets und steht auch im Licht einer Unterscheidung zwischen der Möglichkeit und Praxis des Schreibens und der Benutzung eines Alphabets mit mehr oder weniger beständigen Buchstabenformen. So bilden die alphanumerischen Notationen des Unicode-Standards zwar die Basis für die Festlegung von hunderten Schriftsystemen und schriftverwandten Symbolen für viele digitale Schreibpraxen, dennoch ist Schreiben nicht mit der Verwendung von Buchstaben –

1 Johanna Drucker: *Inventing the Alphabet: The Origins of Letters from Antiquity to the Present*. Chicago: University of Chicago Press 2022, 302.

und schon gar nicht mit der Insistenz auf diskrete Buchstaben- oder Charakterformen in einem strikt-linearen System – gleichzusetzen. Ist, entsprechend Jay David Bolter, »Programmieren [...] das wahre digitale Schreiben«,² so forciert es bestimmte Verhältnisse zwischen Medium, außersprachlichem Bezug und Wissenspraxis. Das soll nicht bedeuten, dass Programmieren das Verständnis des Schreibens radikalisiert, eher, dass spezifische, immer schon inhärente Merkmale des Schreibens, wie es schon N. Katherine Hayles analysiert hat,³ material-spezifisch und graphisch-diskursiv hervorgehoben werden. Mit Bezug auf rezente medienwissenschaftliche Betrachtungen des analogen Schreibens bin ich hier besonders am Zusammenspiel von Stofflichkeit und epistemischem Potenzial des digitalen Schreibens interessiert. Inwiefern können zeitgenössische Diskurse, die der künstlerischen Forschung nahe sind, dazu motiviert werden, über die Art und Weise der Wissensproduktion, -evaluierung und -kommunikation digitalen Schreibens nachdenken?

In der deutschsprachigen Literatur hat vor allem Sybille Krämers Konzeption der Schriftbildlichkeit das Semantikverständnis des Schreibens aufgebrochen, um die operativen Möglichkeiten und Funktionalitäten der Schrift in ihren ikonischen Eigenschaften erkennen zu lassen. Dabei hatte Krämer ursprünglich den Begriff der Schriftbildlichkeit ins interdisziplinäre Spiel gebracht, um das Medienspezifische der Schrift zu erforschen und diese, im speziellen, von der gesprochenen Sprache abzugrenzen.⁴ Schrift ist hier kein graphisches Äquivalent – im Sinne eines neutralen Aufschreibens: Notation – menschlicher Rede, sondern nähert sich der Logik des Bildes an, obwohl diese für Krämer doch eher

2 David Jay Bolter: Digitale Schrift. In: Gernot Grube/Werner Kogge/Sybille Krämer (Hg.): *Schrift: Kulturtechnik zwischen Auge, Hand und Maschine*. München: Wilhelm Fink 2005, 453–467, hier 462.

3 N. Katherine Hayles: *Writing Machines*. Cambridge: MIT Press 2002; Dies.: Print is flat, code is deep: the importance of media-specific analysis. In: *Poetics Today* 25 (2004), H. 1, 67–90.

4 Sybille Krämer: »Schriftbildlichkeit« oder: Über eine (fast) vergessene Dimension der Schrift. In: Dies./Horst Bredekamp (Hg.): *Bild, Schrift, Zahl*. Munich: Wilhelm Fink 2003, 157–176.

eine »operative Bildlichkeit«⁵ besitzt, die mehr Diagrammen und Karten als generell künstlerischen Bildern zu eigen ist. Die Unterscheidung von vermeintlich kategorisch unterschiedlichen Bildlichkeiten ist dabei äußerst umstritten, wobei im besonderen anglophone Ekphrasisliteratur stets die materielle, räumliche and visuelle Untrennbarkeit zwischen der Bildlichkeit von Text und künstlerischen Bildern betont.⁶ Im Folgenden werde ich deshalb versuchen, kurz Krämers Schriftbildlichkeit im Hinblick auf zeitgenössische Literaturdiskurse des *material turn* weiter zu entwickeln, um in deren Kombination das programmierende Schreiben vonseiten einer materiellen Schriftbildlichkeit zu beleuchten. Die Schriftzeichenbenutzung von Codes, die auch so vielem verbalen Schreiben als Algorithmus zugrunde liegt, hat ebenfalls eine äußerst komplexe Relation zur Rede und Schrift und manifestiert, folgt man Philipp Schönthaler, eine epistemische Kraft und Verschiebung des Schreibverständnisses.⁷

In einer Reihe von Aufsätzen hat Sybille Krämer die phonographische Doktrin im konventionellen Verständnis von Schrift und Schriftlichkeit herausgearbeitet. So vergleicht sie zum Beispiel das Schreibverständnis von Gotthold Ephraim Lessing, Susanne Langer und Nelson Goodman. Lessing diagnostiziert im Schreiben eine Kunst der zeitlichen Sukzession (es ist dementsprechend für Narrative des Nacheinander prädestiniert) und beschreibt Malerei als an räumliche

-
- 5 Sybille Krämer: Operative Bildlichkeit: Von der »Grammatologie« zu einer »Diagrammatologie«? Reflexionen über erkennendes »Sehen«. In: Martina Heßler/Dieter Mersch (Hg.): *Logik des Bildlichen: Zur Kritik der ikonischen Vernunft*. Bielefeld: transcript 2009, 94–122.
- 6 Cf. W.J.T. Mitchell: *Composite Art: A Study of the Illuminated Poetry*. Princeton: Princeton UP 1978; Ders.: Spatial form in literature: toward a general theory. In: *Critical Inquiry* 6 (1980), H. 3, 539–567; Ders.: *Iconology: Image, Text, Ideology*. Chicago: University of Chicago Press 1987; Serge Tisseron: All writing is drawing: the spatial development of the manuscript. In: *Yale French Studies* (1994), H. 84, 29–42; Shahar Bram: Ekphrasis as a shield: ekphrasis and the mimetic tradition. In: *Word & Image* 22 (2006), H. 4, 372–378.
- 7 Philipp Schönthaler: *Die Automatisierung des Schreibens & Gegenprogramme der Literatur*. Berlin: Matthes & Seitz 2022.

Gleichzeitigkeit gebunden (d.h. sie ist der statischen Beschreibung des Nebeneinanders zugetan). Ausdruck dieser kontinuierlichen Differenzierungsgeschichte von Wort und Bild als Repräsentation und Präsentation als das Diskursive und das Ikonische ist auch Langers »diskursive[r]« und »präsentative[r] Symbolismus« oder Goodmans mathematisch-anmutende Begrifflichkeit von »syntaktisch dicht« für ikonische und »syntaktisch disjunkt« für diskursive Symbolschemata.⁸ Gemein ist den Ausführungen, die sie kritisiert, dass durch sie Schreiben als räumlich fixierte Rede, als Transkription angesehen wird: Es ist Sprache und nicht Bild. Das bedeutet auch, dass Phänomene wie Zahlensysteme und höhere Mathematik, Logik, Programmier- und formale Sprachen davon ausgeschlossen sind, eine Modalität des Schreibens zu sein.⁹

Krämer hingegen koppelt Schreiben vom phonographischen Zwang ab und manövriert es von linguistischer *Intramedialität* zu *Intermedialität*. Somit wird Schreiben bei ihr nicht länger als Formtransfer von oraler zu visueller Sprache gesehen, was auch ein vermeintliches lineares Fortschreiten aushebelt. Ihr Projekt positioniert Schreiben so nicht nur im Diskursiven, sondern als »hybrid construct in which the discursive and the iconic intersect.«¹⁰ Das »Oszillieren« von Bild und Schrift, »zwischen de[n] Polen« des Diskursiven und Ikonischem¹¹ schreibt allerdings trotz aller Grenzüberschreitungen die vermeintliche Gegensätzlichkeit von beiden ein, im speziellen, wenn Kunstbilder eine (weitere) separate Kategorie darstellen. Polaritäten beschreiben Oppositionalität, auch wenn man eine Linie (Schrift/Bild) zwischen ihnen zieht. Ähnlich verhält es sich auch mit der Logik des Hybrids, das als

8 Sybille Krämer: Sagen und Zeigen. Sechs Perspektiven, in denen das Diskursive und das Ikonische in der Sprache konvergieren. In: *Zeitschrift für Germanistik* 13 (2003), H. 3, 509–519, hier 509; Dies.: Writing, notational iconicity, calculus: on writing as a cultural technique. In: *MLN* 118 (2003), H. 3, 518–537, hier 518.

9 Krämer: Writing, notational iconicity, calculus, 518.

10 Ebd., 519.

11 Krämer: Sagen und Zeigen, 510.

Halbbrut schließlich die Möglichkeit eines Reinheitsgebotes erkennen lässt.

Krämers Ansatz ist so dem von Goodman in *Languages of Art* näher als anfänglich vielleicht vermutet. Dort unterscheidet Goodman autographische Kunst, bei der die Spanne zwischen Fälschung und Original bedeutsam ist, von *allographischer* oder nicht-autographischer Kunst, bei der keine Kopie eines Textes als Fälschung angesehen werden darf und diese als »amenable to notation« gilt.¹² Indem er Schreiben als allographisch bezeichnet, markiert er es als sprachliches Ereignis. Buchstaben sind Einheiten der Schrift, bloße Zeichen von Sprache, in Sprache und beherrschbar durch eine Sprache, die auf Diskursivität reduzierbar ist. Alle graphischen Variationen eines Buchstabens, Symbols oder Charakters sind austauschbar, d.h. zwischen den Allographen eines Zeichens besteht »character-indifference«, Zeichengleichgültigkeit.¹³

Ähnlich bleibt auch für Krämer Schrift immer ein notationales Medium (für Goodman ist es ein Medium mit notationalen Eigenschaften), »welches im Unterschied zum dichten, zum piktoralen Medium mit Lücken bzw. Leerstellen arbeitet.«¹⁴ Für Krämer ist entscheidend, dass die Erkennbarkeit des Zeichens durch Sichtbarkeit zwischenräumlicher Leerstellen und in ihrer »Stellung-innerhalb-einer-Konfiguration«, d.h. durch »Syntax-Visualität« oder »Strukturbildlichkeit«, produziert wird.¹⁵ So ist Goodmans als auch Krämers Schreibverständnis *a priori* notational, weil die Bildlichkeit des meisten Geschriebenen – es sei denn, es handelt sich um als künstlerisch erkannte Bilder(-schrift) – allographisch vordefiniert ist. Betrachtet man Schriftzeichen durch das Ausschlussprinzip eines Stellenwertes, fängt man schon an, diskursiv zu lesen und nicht zu sehen. Ist also die Materialgebundenheit von

12 Nelson Goodman: *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis: New York/Kansas City: Bobbs-Merrill 1968, 121.

13 Ebd., 132; siehe dazu Tilo Reifenstein: Drawing the letter. In: *Drawing: Research, Theory, Practice* 3 (2018), H. 2, 171–186, hier 178f.

14 Krämer: »Schriftbildlichkeit«, 163.

15 Ebd.

Beginn aufgehoben, erkennt man das Schreiben als eher notational und diskursiv.

Ein derartiger Ansatz der Strukturbildlichkeit findet sich auch beim Programmieren wieder: zum Beispiel Syntaxhervorhebungen (*syntax highlighting*), die Wörter- und Zeichenkombinationen in einem Quellcode abhängig von ihrer Bedeutung in unterschiedlichen Farben, Schriftarten und -stilen darstellen. Dies verbessert die Lesbarkeit von Texten für menschliche Leser:innen, lässt Fehler einfacher erkennen, ermöglicht den Überblick über Elemente und Kommentare, hat aber keinen Einfluss auf die Ausführung eines Programmes. Dieser Quellcode ist allerdings auch nicht logisch identisch mit dem Maschinencode des Computers, wie Wendy Hui Kyong Chun es mit Blick auf Speichermaterialität bei der Ausführung von Programmen betont. Die Kompilierung von Quellcode zu Maschinencode ist keine mathematische Übersetzung, »[r]ather, it involves instruction explosion and the translation of symbolic into real addresses.«¹⁶ Die Unterscheidung zwischen unterschiedlichen Schreibenden, Schreibstoffen und Lesenden – jede:r mit materiellem Körper – ist dabei auch beim Programmieren entscheidend.

Ähnlich verhält es sich auch bei *syntax decoration*. Einige Texteditoren erlauben es, bestimmte syntaktische Elemente optisch verändert zu zeigen, z.B. durch Ersetzen eines Zeigeroperators (->) durch ein tatsächliches Pfeilsymbol (→) oder durch Austauschen von Hinweisen zur Textdekoration wie kursiv, fett oder unterstrichen in Quellcodekommentaren durch eine tatsächliche *kursive*, **fette** oder unterstrichene Hervorhebung. Die Terminologie der Dekoration deutet wiederum auf unterschiedliche Leser:innen, aber auch auf eine vermeintliche Bedeutungslosigkeit hin. In der *Überschreibung* in den Maschinencode wird der dekorierende Quellcodekommentar von Signifikanz befreit. Jedoch ist dieses Lesen wie jedes andere auch an vorgegebene Paradigmen gebunden. So beschreibt Vilém Flusser, dass sich alles Lesen – selbst das

16 Wendy Hui Kyong Chun: *Programmed Visions: Software and Memory*. Cambridge/London: MIT Press 2011, 22.

von Erbsen oder solches, das Hühner betreiben – an bestimmte Maßstäbe bindet und so auch interpretierend vorgeht.¹⁷ Lesen impliziert, dass Inhalte chiffriert vorhanden sind und dass diese hervorgeholt, d.h. entziffert werden können. Das Geschriebene wiederum hat somit immer mehr als nur eine:n Leser:in. Michael Mateas und Nick Montfort beschreiben dies ähnlich in ihrer Diskussion bezüglich der notwendigen Kompromisse, menschliche und computerisierte Lesbarkeit sicherzustellen: »All coding inevitably involves double-coding. ›Good‹ code simultaneously specifies a mechanical process and *talks about* this mechanical process to a human reader.«¹⁸

Das Verständnis von Visualität mit Fragen der Syntax zu verknüpfen, sei es wie bei Krämer oder wie bei der Ornamentierung des Quellcodes, wirkt jedoch recht zurückhaltend, besonders im Vergleich zu jener zeitgenössischen Literatur, die mit künstlerischer Forschung liebäugelt oder dem *material turn* verpflichtet ist. So zeigt Anne M. Royston in *Material Noise: Reading Theory as Artist's Book* wie typographische (Un)Lesbarkeit, Farbe, Bild, Format und Formatierung sowie Papier, Bindungen und Ähnliches nicht nur Teil des Repertoires der Signifikation in Künstlerbüchern sind, sondern auch bei Theorieschreiben und -lektüre wirksam werden.¹⁹ Roystons Argument macht aus den vermeintlich nicht-semantischen Materialien des Schreibens nicht nur einen Störfaktor, sondern eine notwendige Kontingenz für die Deutung und Bedeutsamkeit des Geschriebenen. Während Goodmans und Krämers allographische Schriftlichkeit durch die Materialität des Schreibens hindurchliest, wird es bei Royston Teil einer komplexeren, multivalenten

17 Vilém Flusser: *Die Schrift. Hat Schreiben Zukunft?* Hg. von Andreas Müller-Pohle. Göttingen: European Photography 52002 [1987], 77–90.

18 Michael Mateas/Nick Montfort: A box, darkly: obfuscation, weird languages, and code aesthetics. In: *Digital Arts and Culture (DAC) Conference: Digital Experience: Design, Aesthetics, Practice*. Copenhagen 2005, 1–10, hier 9. <https://users.soe.ucsc.edu/~michaelm/publications/mateas2-dac2005.pdf> (07.01.2023); Hervorhebung im Original.

19 Anne M. Royston: *Material Noise: Reading Theory as Artist's Book*. Cambridge/London: MIT Press 2019.

Argumentation, in der »what is usually seen as nonproductive can nonetheless produce meaning.«²⁰ Oberfläche, Style, Ornament und vergleichbares Vokabular müssen bei Royston neu definiert werden, um ihre semantische Kraft zu schüren. Dass Royston auch Hayles aufgreift, um die hypertextuelle Fragmentier-, Rekombinier- und Transformierbarkeit von bestimmten gedruckten Büchern und deren Argumentationen aufzuzeigen, bestätigt die rekursive Les- und Sehbarkeit von Geschriebenem allgemein durch spezifische stoffliche Affordanzen.

Eine vergleichbare, materiell aufmerksame Lektüre unternimmt Juliet Fleming in ihrem Buch *Cultural Graphology*.²¹ Indem sie Druckfehler, Schriftverzerrungen, die Praxis des Zerschneidens von Drucksachen mit Schere und Messer sowie schwarz gedruckte Seiten untersucht, schreibt Fleming die Druckkultur des frühneuzeitlichen Englands neu und liest, was normalerweise als extrasemantisch entweder *übersehen* oder *nur* gesehen wird. In Anlehnung vor allem an Jacques Derridas *De la grammatologie* geht sie der Literalität der Lettern nach, ähnlich wie Ben Kafka, der durch eine materialverbundene Analyse der Schreibarbeit die bürokratische Gewalt in der Schreibkultur der Französischen Revolution auffindet.²²

Dabei ist die Materialitätsbetrachtung unbedingt an eine Diskursbetrachtung gebunden, wie es auch Marjorie Perloff in *Infrathin* zeigt und dort betont, dass »[t]he attention to difference, to the infrathin, is especially important [...] in the Age of Digital Reproduction.«²³ Auch bei ihr sind Bildlichkeit und Materialität keine Frage künstlerisch optionaler Interpretation, die Diskursivität und Wissensproduktion im Sinne einer Körper-Geist-Trennung außen vorlässt. Die Termini stehen in dezentrierter Relation ohne Grenzpatrouille zueinander. So zeigt Perloff,

20 Ebd., 16.

21 Juliet Fleming: *Cultural Graphology: Writing after Derrida*. Chicago; London: University of Chicago Press 2016.

22 Kafka, Ben: *The Demon of Writing: Powers and Failures of Paperwork*. New York: Zone Books 2012.

23 Marjorie Perloff: *Infrathin: An Experiment in Micropoetics*. Chicago: University of Chicago Press 2021, 26.

dass Layout mit Klang korreliert (*co-relāt-*), der mit Etymologie korreliert, die mit Grammatik korreliert, die mit Bildlichkeit korreliert ...

Das *Signifikante* ist so nicht mit dem Symbolischen oder der Referenz gleichsetzbar. Beide rücken den Körper des Materials immer schon in Richtung einer Repräsentation, bei der er verloren geht. Wir schreiben, sehen und lesen (in und mit dem) Körper, ebenso wie Maschinen. Das Material produziert seine eigene Seh-, Les-, Hör- und Fühlbarkeit, die sich, wie es Craig Dworkin auch in seinem neusten Buch *Radium of the Word* zeigt, selbst miteinschreibt.²⁴ So findet Dworkin in der vermeintlichen Oberfläche von Texten und ihrer Zeichensetzung, und selbst in Eigennamen und zwischen Sprachen, bild-schrift-materielle Resonanzen, die nicht auf Diskursivität, Ikonizität oder Materialität allein beschränkbar sind. In diese Reihe gehören auch: Gregory Ulmers *Applied Grammatology*, Manuel Portelas *Scripting Reading Motions*, Lori Emersons *Reading Writing Interfaces*,²⁵ wobei hier vor allem neuere und im deutschsprachigen Diskurs vielleicht weniger bekannte Texte behandelt werden.

Die Aufzählung dieser Veröffentlichungen soll nicht nur ein paar Details der enormen Fülle der rezenten Literatur kennzeichnen, sondern vor allem unterstreichen, dass eine Analyse des (digitalen) Schreibens und seiner Schriftlichkeit nicht nur medienspezifisch, sondern auch materialspezifisch vorgeht, dass Schreiben selbst Wissensproduktion sein kann und nicht auf deren Erfassung, Speicherung und Übermittlung als Kommunikations- oder Repräsentationsapparat reduzierbar ist, und dass dieses epistemische Potenzial des Schreibens mit dem Körper des Schreibens und Lesens korreliert. Krämers intermedialer Ansatz erlaubt es ihr, ein fälschliches Linearitätsgebot der Schrift als stete, nacheinander gereihete Folge von Schriftzeichen zu kritisieren:

24 Craig Dworkin: *Radium of the Word: A Poetics of Materiality*. Chicago: University of Chicago Press 2020 (= Thinking Literature, hg. von Nan Z. Da/Anahid Nersessian).

25 Gregory L. Ulmer: *Applied Grammatology: Post(e)-Pedagogy from Jacques Derrida to Joseph Beuys*. Baltimore/London: Johns Hopkins UP 2019; Manuel Portela: *Scripting Reading Motions: The Codex and the Computer as Self-Reflexive Machines*. Cambridge: MIT Press 2013; Lori Emerson: *Reading Writing Interfaces: From the Digital to the Bookbound*. Minneapolis: University of Minnesota Press 2014.

»the belief in the one-dimensional nature of the written image as a linear series of letters disregards the fact that every written text uses the *two-dimensionality of surfaces*.«²⁶ Jedoch legt ein oberflächlicher Bildlichkeitsbegriff die materielle Dimension des Schreibens als *image* und nicht als *picture*, wie es Royston oder Perloff verstehen, aus. Schreiben besitzt immer auch ein Substrat, das ihm nicht nur zugrunde liegt, sondern das substantiell die Eigenschaften als korrelierender Körper mitschreibt. So betont Annette Vee in Hinblick auf die Programmierkompetenz: »[t]he historically persistent trade-offs between the roles of hardware and the many layers of software point to the difficulties of drawing lines between the physical engineering of computational machines and the many different languages we use to control them.«²⁷ Auch wenn Java in den 1990er Jahren eine praktisch plattformunabhängige Programmierumgebung anbot, legt der Gebrauch von Programmiersprachen wie Lisp für Vee nahe, dass sich Sprachspezifikationen unweigerlich mit sozialen und materialtechnischen Faktoren überschneiden.

Der enorme Sinn(es)reichtum, der in der aufgeführten Literatur zum Ausdruck kommt, könnte in gewissem Gegensatz zu den Übersetzungsbemühungen des zuvor beschriebenen Texterkennungsprozesses stehen. Allerdings ist die digitale Schriftlichkeit immer auch an eine materielle Basis gebunden und ihre Ikonizität bleibt, zumindest virtuell, bestehen. Somit kombiniert digitales Schreiben pluriforme (nicht omni- oder amorphe) Bildlichkeit mit Formalisierung. Unicode standardisiert und vereinigt hunderttausende Zeichen in einem Zeichensatz als *Universal Coded Character Set*. Jedem Zeichen ist ein sogenannter *code point* zugeschrieben, das heißt, alle Allographen, Glyphen oder sonstigen Variationen haben den gleichen Verweis, womit Zeichen sortiert und normalisiert werden. Der Unicode-Zeichensatz bietet eine Standardentsprechung zwischen den Binärzahlen der elektronischen Prozessierbarkeit, die Computer verstehen, und lesbaren Zeichen. Die Normalisierungsbedingungen des Unicodes fasst Julia Nantke mit der

26 Krämer: *Writing, notational iconicity, calculus*, 520; Hervorhebung im Original.

27 Annette Vee: *Coding Literacy: How Computer Programming Is Changing Writing*. Cambridge; London: MIT Press 2017, 113.

Feststellung zusammen, dass »[d]ie unauflösliche Verschränkung von Schriftbild und Textsemantik, die seit dem *material turn* in den Fokus der literaturwissenschaftlichen Aufmerksamkeit gerückt ist, [...] also gerade kein Teil digitaler Schriftlichkeit« ist.²⁸ Jedoch hat auch Programmieren eine Schriftbildlichkeit (wenn auch kein singuläres Schriftbild), und Digitalität ist immer auch materialgebunden. Das Schriftbild des programmierenden Schreibens ruft binärkodale die Entwicklungsgeschichte des Programmierens von elektromechanischen Schaltkreisen (an/aus) zu symbolischer Repräsentation wach. So beschreibt Bolter das Programmieren sowie das Lesen und Schreiben des Computers allgemein als »exercise in applied semiotics«.²⁹ Innerhalb des geschlossenen Systems (Stromkreises) verweist (erst eine Verdrahtung) ein Zeichen auf ein anderes. Damit gewinnt eine Anreihung von Zeichen durch intern definierte Begriffe und Verfahren an Bedeutung. Manche Systeme, die sich auf neuronale Netze stützen und stochastische Modelle benutzen, mögen sich zwar von diesem »solipsistischen ›Symbolkarussell‹« weg- und zur Erdung hinbewegen.³⁰ Wie welche Sensoren Daten gewinnen, besteht als Frage jedoch immer fort.

Bernard Stiegler's Konzeption der Grammatisierung bietet hier ein produktives Prinzip, um die Wechselverhältnisse von nicht-digitalem Schreiben und programmierendem Schreiben, oder auch weitergefasst, von Sprache und Schreiben und Code zu theoretisieren. Stiegler appliziert Rationalisierungstheorien industrieller Prozesse in Hinblick auf die symbolische Repräsentation von Schreiben und Programmieren: »La grammatisation est la production et la discrétisation de structure«.³¹ Die

28 Julia Nantke: Normalisierung als Bedingung von Schriftlichkeit am Beispiel digitaler Repräsentationen von Schrift. In: Martin Bartelmus/Alexander Nebrig (Hg.): *Schriftlichkeit. Aktivität, Agentialität und Aktanten der Schrift*. Bielefeld: transcript 2022, 39–54, hier 43.

29 J. David Bolter: *Writing Space: The Computer, Hypertext, and the History of Writing*. Hillsdale: L. Erlbaum 1991, 203.

30 Hannes Bajohr: Dumme Bedeutung. Künstliche Intelligenz und artifizielle Semantik. In: *Merkur* 76 (2022), H. 882, 69–79, hier 75.

31 Bernard Stiegler: *Mécréance et Discrédit (I): La Décadence des Démocraties industrielles*. Paris: Galilée 2004, 114.

Grammatisierung der Industriemaschine liegt in ihrer Diskretisierung der Gesten der Arbeit, die der Schrift in der graphischen Diskretisierung von Sprache durch Sätze, Zeilen, Worte, Buchstaben und Charaktere. Der Prozess der Grammatisierung eröffnet bestimmte Affordanzen, d.h. er bedingt und ermöglicht Eigenschaften wie Reproduzierbarkeit, Distribution und Anwendungsumfang. Nantkes Normalisierung, Druckers »graphically discrete elements« sowie Goodmans »syntactic requirements of disjointness and of finite differentiation«³² (die Krämer weiterverwendet) finden in Stieglers Grammatisierung ebenso Platz. Als transformativer Prozess rationalisiert digitales Schreiben Elemente der analogen Praxis, die bestimmte epistemische Propositionen des programmlichen Schreibens zwischen Bild, Schrift und Material formulieren lassen. Diskretisierung und Sequentialität bescheinigen eine Fragmentier-, Rekombinier- und Transformierbarkeit des digitalen Schreibens – wie es auch dem Hypertext eigen ist, – die bei Fehlerfreiheit eine Transmission von Daten versichern. Die Einschreibung von hochgradig vorbestimmten Abläufen ordnet die Lesung und Ausführung von Zeichen, zumindest für die Maschine, kategorisch (an/aus) und hierarchisch (*script*). Digitales Schreiben bietet somit eine Vorlage für explizite und prozedurale Denkweisen und damit verbundene Möglichkeiten, neue Arten von Wissen zu generieren.

Ich habe mich soweit beim Schreiben dieses Beitrags auf zeitgenössische kulturwissenschaftliche Literatur bezogen, die sich vor allem an Derridas Annäherung an das Schreiben lehnt. Stieglers Grammatisierung z.B. verwertet Aspekte von Derridas Grammatologie weiter, um seine kritische Aufmerksamkeit auf die Untersuchung der konstitutiven Prozesse, die den sogenannten Menschen und Technologien *gemeinsam* zugrunde liegen, zu lenken, und anderswo anhaltendes Gegensatzdenken zwischen den beiden zu umgehen. Ulmer wiederum sah in Derridas Schreibprozess die Entwicklungsmöglichkeiten einer Art des Schreibens und pädagogischen Praxis, die geisteswissenschaftliche Forschung und Lehre angesichts vernetzter audiovisueller Medien dienlich sei. Flemings *cultural graphology* unternimmt eine grammatologische Arbeit,

32 Goodman: *Languages of Art*, 137.

die in ihrer physischen Materialspezifität so nur von Derrida vorgeschlagen wurde. Derrida refokussiert sprachwissenschaftliche Arbeit »von der Semiologie zur Grammatologie, d.h. von einer Wissenschaft der (sprachlichen) Bezeichnung zu einer [...] des schriftsprachlichen Zeichens«. ³³ Dieses Zeichen ist bei Derrida nicht auf Signifikation reduzierbar. Es wird gesehen-gelesen und hat Körper. So scheint es zum Abschluss naheliegend, noch einmal zu Derrida zurückzukehren, um in zwei frühen Texten bildmaterialdiskursive Anstöße für eine Betrachtung der epistemischen Eigenschaften digitalen Schreibens zu finden.

Vor allem in *De la grammatologie* erörtert Derrida die Herausforderungen, die an eine (unmögliche) *science de l'écriture* gestellt sind. Die grundsätzlichen Probleme einer solchen Disziplin umfasst

»[...] l'articulation d'une graphie individuelle et d'une graphie collective, du ›discours‹, si l'on peut dire, et du ›code‹ graphiques, considérés [...] du style et de la connotation ; quant à l'articulation des formes graphiques et des diverses substances, des diverses formes de substances graphiques (les matières : bois, cires, peau, pierre, encre, métal, végétal) ou d'instruments (pointe, pinceau etc. etc.) ; quant à l'articulation du niveau technique, économique ou historique (par exemple au moment où s'est constitué un *système* graphique et au moment, qui n'est pas nécessairement le même, où s'est fixé un *style* graphique) ; quant à la limite et au sens des variations de styles à l'intérieur du système ; quant à tous les investissements auxquels est soumise une graphie, dans sa forme et dans sa substance.« ³⁴

Ähnlich wie in Druckers alphabetgeschichtlicher Argumentation ist Schreiben für Derrida weder auf eine spezifische Symbolverwendung noch auf eine gewisse mediale Instanziierung reduzierbar. Schreiben ist materielle Praxis, egal ob als Spur im Schnee oder elektrographische

33 Birgit Mersmann: *Schriftikonik. Bildphänomene der Schrift in kultur- und medienkomparativer Perspektive*. Paderborn: Wilhelm Fink 2015, 24.

34 Jacques Derrida: *De la Grammatologie*. Paris: Éditions de Minuit 1967, 132f.; Hervorhebung im Original.

Verarbeitung von Spannungswerten im Logikgatter. Bedeutungsabsicht und Denotation werden relegiert, um sich auf die *Graphie* selbst zuzubewegen und deren Fortschreiten und ihre Einschreibungen zu erforschen. So ist es für Derrida interessant, wie sich graphische Formen etablieren. Durch die Unterscheidung der Konstitution eines graphischen Systems von seiner Fixierung verweist Derrida auf historische Umwandlungsprozesse, die Avancen mit der Möglichkeit von historischer Originarität unterwandern. Schreiben wird so abseits von Inhärenz und Idealform in seinem sozialen, politischen und materiellen Gefüge kontextualisiert. Vee beschreibt so zum Beispiel unter Verweis auf Sherry Turkles Forschungen, dass die steigende Popularität von strukturierter, objektorientierter Programmierung in COBOL (Common Business Oriented Language) und ALGOL (Algorithmic Language) in den 1960er Jahren wegen ihrer besseren Kollaborationsmöglichkeiten durch größere Modularität auch eine Stilstarrheit mit sich brachte, die mehr Männer als Programmierer anzog.³⁵ Die komplizierte Verflechtung wirtschaftlicher, technologischer und sozialer Faktoren verstärkt die Notwendigkeit, das Schreiben im Hinblick auf solche Merkmale zu betrachten, anstatt es als etwas Gegebenes zu sublimieren. Schreiben ist also auch eine Frage der Investitur: Wie wird es wann durch wessen Anliegen (*Anlagen*) eingesetzt?

Derridas Schreiben ist nicht auf Kommunikationsbestrebungen reduzierbar und so nicht durch Notationsansprüche vordefiniert. Stattdessen ermutigt er eine Aufmerksamkeit, die man bei der Analyse von Kunstwerken erwarten würde, um Schrift materiell untersuchbar zu machen. Syntax und Semantik werden als flexibel, irreduzibel und vielfältig verstanden, sodass der anthropozentrische Schwellenunterschied von Interpretation und Ausführung selbst infrage gestellt wird. Wer schreibt/liest hier mit welchen Maßstäben? Derridas parenthetische Aufzählung von Substanzen und Instrumenten verdeutlicht, dass Form und Substanz der *Graphie* nicht als Polaritäten agieren, nicht koextensiv sind, weil es zwei kategorische Werte betrifft, sondern umfassend und dezentriert die Variabilität des Schreibens bekunden. Silizium,

35 Vee: *Coding Literacy*, 109f.

Kupfer und Plastik eines Halbleiters sind so notwendiger Teil einer Untersuchung, die die unmöglichen Grenzen des Schreibens erörtert.

Die Umschreibung von Geschriebenem durch OCR und ähnliche Schrifterkennungsverfahren, die hier anfänglich als exemplarisch für ein divergentes Schreibverständnis angerissen wurde, entzieht sich somit dem floskelhaften Verweis der Reduktion des Schreibens und wird als elektrographische Übersetzung oder Grammatisierung einer kulturspezifischen Material- und Stilphase der *Graphie* zuordenbar. Schreiben als Kulturpraxis, wie es auch Krämer betrachtet, muss als materielle, epistemische Prozesshaftigkeit des Spätkapitalismus untersucht werden. Die Rationalisierungstendenzen, die man in der digitalen Schrift diagnostizieren möchte, sind eingebettet in global-lokale Extrahierungs- und Ausbeutungsprozesse des frühen 21. Jahrhunderts. Schreiben ist kein absolutes Phänomen, sondern immer auch spezifisch instanziiert.

Derridas Grammatologie trägt die Spuren seiner vorhergehenden Erkundungen des Schreibens und der Sprache, insbesondere seiner ausführlichen Einleitung zu Edmund Husserls *Frage nach dem Ursprung der Geometrie*. In der Einführung konstatiert er Husserls Bedürfnis nach einer sprachlichen Eindeutigkeit und Genauigkeit, die durch lineare und unabänderliche Ausdrucksformen erreicht werden soll. Ein frei-assoziatives Sprachspiel gefährdet jedoch stets die Deutbarkeit und Deutlichkeit des Schreibens. Bedeutung kann weder für die Schreibenden noch für die Leser:innen festgelegt werden. Derrida beleuchtet so zwei Ansätze in der Beziehung von Sprache und Geschichte. Der eine, Husserls, pocht auf eine Sprache der Eindeutigkeit und Transparenz, deren Gebrauch historisch bestimmbar, rein und originär ist; der andere

»[...] ressemblerait à celle de][ames] Joyce répéter et reprendre en charge la totalité de l'équivoque elle-même, en un langage qui fasse affleurer à la plus grande synchronie possible la plus grande puissance des intentions enfouies, accumulées et entremêlées dans l'âme de chaque atome linguistique, de chaque vocable, de chaque mot, de chaque proposition simple, par la totalité des cultures mondaines,

dans la plus grande génialité de leurs formes (mythologie, religion, sciences, arts, littérature, politique, philosophie etc.) [...].³⁶

Joyces Werk signalisiert für Derrida eine beispielhafte, aber unausweichliche Fissur im vermeintlich grundlegenden Darstellungswert der Sprache und jeglicher festen Entsprechung von Sprache und Welt. Husserls Anspruch auf Eindeutigkeit und Genauigkeit wird in *Finnegans Wake* – noch mehr als in *Ulysses* – demonstrativ und stetig durch eine Reihe von grammatischen, etymologischen und narrativen Strategien, aber auch durch die notwendige Zuflucht in schriftliche Sprache an sich unterwandert. Nicht jede:r ist Joyce, und ebenso ist nicht jedes linguistische Atom von einem *auteur* steuerbar. Schreiben ist Teilhabe an einer historischen Praxis, deren Geschichte und Eigenschaften als Absolut unergründbar bleiben. Als individuelle Privatsprache verbliebe sie unkommunikativ, als Kulturpraxis überschreitet sie die Kontrolle und Beherrschbarkeit im Einzelnen. Wissenstransmission, wie man es von Sprache als Hervorbringerin einer Kommunikationssituation erwarten würde, bedingt die Kontinuität von Bedeutung zwischen Personen und über Zeiträume. Diese Eindeutigkeit («l'impératif de l'univocité») – oder Eineindeutigkeit, wenn man Notationsansprüche an sie stellt – wird immerfort durch das Schreiben einer Äquivozität (gar einer »équivocité radicale«) verfolgt.³⁷ Auch andersherum manifestiert *Finnegans Wake* kein äquivokales Werk, da es Deutbarkeit und Deutlichkeit nicht radikal aufgeben kann. Der Anspruch an Codes/Diskretisierung entbindet ihn nicht von Uneindeutigkeit. Auf Grund des hohen Diskretionsgrades von Code können komplexe, verkettete und mehrschichtige Prozesse geschrieben und von Computern ausgeführt werden. Diese Prozesse können in miteinander verbundene Operationen verschachtelt werden

36 Jacques Derrida: Introduction. In: Edmund Husserl: *L'origine de la géométrie*. Paris: Presses Universitaire de France ⁶2010 [1962], 3–171, hier 104.

37 Derrida: Introduction, 101, 104. Äquivozität ist hier Zwei- oder Mehrdeutigkeit vorzuziehen, da es den Verweis auf Gleichheit (equal, égal, egal) des Zitats bewahrt und nicht auf Zählbarkeit ummünzt.

oder Schlaufen und andere Formen durchlaufen. Die Kopie eines Ablaufs kann Teil einer Programmbibliothek oder direkt in ein anderes Programm eingefügt werden. Jedoch ist die Äquivalenz der Ergebnisse des Codes im neuen Zusammenhang nicht garantiert. Ebenso sind die sensorielle Erschreibung von Daten und mögliche Fehlersituation wie Signalverzögerungen oder spezifische physikalische Fehler, die durch immer höhere Integrationsdichten der Halbleiterprozesse versucht sind, auch für digitales Schreiben konstituierend.

Zugleich tritt das epistemische Moment des Schreibens hier wieder spürbar in den Vordergrund. Derridas Einleitung zeigt, dass Husserl nicht nur das Schreiben benötigte, um die Beständigkeit idealer Objekte zu gewährleisten, sondern dass er tatsächlich auch das Schreiben als Garant dafür begriff, dass es sich tatsächlich um ideale Objekte handelt. Elektrographisches Schreiben konstituiert ebenso eine Welt im Sinne ihrer Logik. Mark Marino unterstreicht mehrfach die Wichtigkeit, Code auch durch dessen Verhältnisse zu sozialen, politischen und wirtschaftlichen Machtstrukturen zu verstehen.³⁸ So muss die determinative und hierarchische Prozesshaftigkeit des Ordnen und Systematisierens, die dem digitalen Schreiben zu eigen ist, gerade mit Blick auf andere Formen des Wissens und deren Organisation hinterfragt werden, um zu erörtern, wie Schrift, Bild und Material ihre Welt formiert. Eine Betrachtung des digitalen Schreibens, losgelöst von seinen bildmaterialdiskursiven Bestandteilen – *It's in the cloud!* –, wäre ein Fehlkalkül, das nicht nur dem bedingungslosen Verschleiß von seltenen Erden und enormen Energiehunger seiner Systeme schulterzuckend gegenübersteht, sondern auch die operativen Gesetzmäßigkeiten außerhalb der Praxis von einem breiteren Verständnis von Schrift sucht.

38 Mark C. Marino: *Critical Code Studies*. Cambridge/London: MIT Press 2020.

Maschinenschreiben zwischen Energieia und Ergon

Eine sprachphilosophische Untersuchung von Large Language Models

Lucas Falkenhain

1. Einleitung

Rund ein halbes Jahrhundert nach Joseph Weizenbaums Chatbot ›ELIZA¹ markiert die Aneignung der Sprache durch *Large Language Models* wie OpenAIs ChatGPT und GPT-4 einen epochalen Wendepunkt, der nicht nur die Macht der Sprache und ihre Auswirkungen auf das menschliche Denken und Handeln erneut unter Beweis stellt. Er aktualisiert vor allem die Frage nach der Möglichkeit (starker) künstlicher Intelligenz, die – den Fiktionen des Silicon Valley folgend – durch

1 Das 1966 veröffentlichte Programm, das ein Gespräch mit einem Psychotherapeuten simulierte, gilt als das erste Sprachmodell, das in breiter Öffentlichkeit mit potenziell ernst zu nehmender künstlicher Intelligenz in Verbindung gebracht wurde. Vgl. Joseph Weizenbaum: ELIZA. A Computer Program For the Study of Natural Language Communication Between Man And Machine. In: *Communications of the ACM* 9/1 (1966), 36–43. Weizenbaum kommentiert: »Einige Personen waren nur sehr schwer davon zu überzeugen, dass ELIZA [...] nicht menschlich ist.« (Vgl. Joseph Weizenbaum: ELIZA, 42. Übers. d. Verf.); Nehlsen, Lukas: Chatten mit Nirgendwo? Der Leib als Ausdruck und die Sprache der KI. In: Martin W. Schnell/Lukas Nehlsen (Hg.): *Begegnungen mit Künstlicher Intelligenz. Intersubjektivität, Technik, Lebenswelt*. Weilerswist: Velbrück 2022, 120–122.

die Entwicklung sprachlicher Kompetenzen und Interfaces womöglich schon bald die Schwelle des reinen Werkzeugs überschreiten, ja vielleicht sogar die Demokratisierung von Sprache und Maschine einfordern könnte. So spekulativ derartige Narrative auch sein mögen, so wecken sie doch einige Fragen über den Verbleib der Sprache ur-eigenen Agentialität.² Die genuin anthropozentrisch geprägte Sprach- und Schriftlichkeitstheorie sieht sich inzwischen dem Feld einer sich ausdehnenden Mensch-Computer-Interaktion sowie der diese begleitenden pragmatischen Kommunikationstheorie gegenübergestellt.

Diese Spannungen möchte ich im Folgenden aus sprachtheoretischer Perspektive reflektieren und das Phänomen ›intelligenter‹ schreibender Maschinen im Kontext des Sprachorganismus Wilhelm von Humboldts situieren. Ausgehend von der Unterscheidung der Sprache in die zwei wesentlichen Momente *Energeia* (Tätigkeit) und *Ergon* (Werk) drängt sich die Frage auf, inwiefern das maschinelle Schreiben jüngerer Chatbots als *agentielle* Sprachtätigkeit gewertet werden könnte.³ Begründet die Veräußerung der menschlichen Sprache bereits das Denken in Sprache? Sind schreibende Maschinen dazu in der Lage, durch sprachliche Referenz auf die Welt eine eigene Semantik, vielleicht sogar eine virtuelle Individuation zu entwickeln? Oder bedeutet die logisch algorithmisierte Textverarbeitung lediglich die retrospektiv perpetuierende, statistische Reproduktion menschlicher Schriftkultur? Verbirgt sich hinter intelligenten Chatbots somit also doch nur ein simulatives Täuschungsmanöver neuer Algorithmen?

2 Zum Terminus der *Agentialität* siehe Martin Bartelmus/Alexander Nebrig: *Schriftlichkeit. Aktivität, Agentialität und Aktanten der Schrift*. Bielefeld: transcript 2022.

3 Zum Terminus *agentiell* siehe Karen Barad: *Agentieller Realismus. Über die Bedeutung materiell-diskursiver Praktiken*. Übersetzt von Jürgen Schröder. Berlin: Suhrkamp 2012.

2. Sprachdenken und Logik

Large Language Models stehen in der Tradition der logischen, elektronischen Datenverarbeitung, die seit den informationstheoretischen Gehversuchen in der ersten Hälfte des 19. Jahrhunderts immer wieder bemüht wird, um die menschliche Sprache nach mathematischen Prinzipien erst zu modellieren und anschließend zu synthetisieren. Trotz des hiermit einhergehenden offensichtlichen Mangels an Geist und Körper scheint es zeitgenössischen Modellen wie etwa GPT-4 zu gelingen, natürliche Konversationen zu führen, logische Urteile zu fällen oder sich emotional in Menschen hineinzusetzen. Fraglich bleibt jedoch, ob das logische Kalkül der Maschine jenseits seiner Phänomenologie lexikalisch vergleichbarer Schrifterzeugnisse überhaupt dazu in der Lage ist, ein Denken in Sprache zu entwickeln. Denn der Veräußerung von Sprache geht auch bei Humboldt als unhintergehbare Bedingung stets das Denken in Sprache voraus. Im Folgenden soll dieses Verhältnis von Denken und Sprache bei Humboldt näher untersucht und dargelegt werden, warum sich die maschinelle Schrifterzeugung nur als *Ergon* jenseits einer anthropozentrischen Geistestheorie verorten lässt.

Um die Bestandteile der humboldtschen Sprachtheorie darstellen zu können, muss jedoch zunächst eingeschränkt werden, dass sich die dialektischen Elemente seiner Sprachanthropologie nur als interdependenten Organismus erschließen lassen. Einheiten wie der Mensch, die Welt, seine Sprache und auch das Denken sind demnach nicht als distinkt voneinander zu trennende Einheiten, sondern nur als einander wechselseitige zugewandte Teile einer organisch verwachsenen Sprachanthropologie zu verstehen.⁴ Demzufolge liefert Humboldt auch keine engere Defi-

4 Vgl. Gerda Haßler: Die These von der Sprachrelativität des Denkens in der Aufklärung und bei Wilhelm von Humboldt. In: Klaus Welke (Hg.): *Sprache, Bewußtsein, Tätigkeit. Zur Sprachkonzeption Wilhelm von Humboldts*. Berlin: Akademie-Verlag 1986, 72–176; Clemens Menze: *Wilhelm von Humboldt. Lehre und Bild vom Menschen*. Ratingen: Henn 1965, 224; Tilman Borsche: Humboldts These von der Individualität der Einzelsprache. Metapher oder Terminus? In: Ute Tintemann/Jürgen Trabant (Hg.): *Wilhelm von Humboldt. Universalität und Individualität*. München: Fink 2012: 88–90; Isabella Ferron: *Sprache ist Rede. Ein Beitrag zur*

inition des Begriffs ›Denken‹, sondern nimmt es als apriorische Prämisse einer Theorie, in der er vielmehr die unmittelbare Abhängigkeit des Denkens von der Sprache beleuchtet. So beschreibt Humboldt, dass die Möglichkeit des Denkens nur in und durch die Sprache selbst vorstellbar ist: Ohne die Sprache »ist die Bildung des Begriffs, mithin alles wahre Denken, unmöglich.«⁵ Die Veräußerung der Sprache ist in dieser Konsequenz somit nicht weniger als der unmittelbare »Abdruck des Geistes«⁶.

Wie steht es dem gegenüber um die Möglichkeit eines maschinellen Denkens in Sprache? Kann maschinelles Sprachdenken überhaupt existieren, wenn die innere Sprachform von Computern lediglich auf symbolische Programmiersprachen und damit auf eine binäre Genetik zurückzuführen ist? An Stelle der von Natur gegebenen individuellen Geisteskraft,⁷ die eine Vorstellung durch einen abstrakten Begriff zu umreißen versucht, tritt nämlich das Prinzip des Prozessierens diskreter Zahlen. Deren Gesetz wiederum, die mathematische Logik, erlaubt weder Leerstellen, noch kennt sie die semantische Unschärfe der Sprache, die im menschlichen Denken kontinuierlich durch die wechselseitige Verständigung bearbeitet werden muss.⁸ Das maschinelle Prozessieren vollzieht sich vielmehr über syntaktische Ge-

dynamischen und organistischen Sprachauffassung Wilhelm von Humboldts. Würzburg: Königshausen & Neumann 2009, 113–141.

- 5 Wilhelm von Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues. In: Albert Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 6. Berlin: Behrs 1907, 155.
- 6 Wilhelm von Humboldt: Über den Dualismus. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 6, 23.
- 7 Die Kraft des Geistes versteht Humboldt als eine von Gottfried Wilhelm von Leibniz entlehnte Idee, die als innere Veranlagung des Menschen sämtliches Denken und Handeln (in Sprache) motiviert. Vgl. Menze: *Wilhelm von Humboldt*, 96–105.
- 8 Bedeutung erschließt sich bei Humboldt als fluides Gebilde, welches aufgrund divergenter Weltwahrnehmungen (subjektive Weltansichten) im jedesmaligen Sprechen neu verhandelt und rückversichert werden muss. Vgl. Wilhelm von Humboldt: Grundzüge des allgemeinen Sprachtypus. In: Albert Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 5. Berlin: Behrs 1906, 396: »Alles Verstehen ist daher immer zugleich ein Nicht-Verstehen [...]«

setzmäßigkeiten, die als mathematische Relationen der Berechnung statistisch wahrscheinlicher Sprache verhelfen. So werden Aussagen in berechenbare Texteinheiten (*Token*) zerlegt, deren Gewichtungen im Entscheidungspfad möglicher Wortfolgen kontinuierlich durch numerische Multiplikatoren in den neuronalen Schichten verstärkt oder abgeschwächt werden.⁹ Infolge dieser stochastischen Berechnungen werden auch bei GPT-Modellen Wortbedeutung und -kontext weder intrinsisch auf ihre Konnotationen hin reflektiert, noch in Beziehung zu einer Sprachgemeinschaft.

Entscheidend für eine Bewertung der Möglichkeit des maschinellen Sprachdenkens ist jedoch die ausbleibende synergetische Verbindung von Denken und Sprachveräußerung, da bei der maschinellen Verarbeitung keine unmittelbare Verschränkung zwischen der inneren (gedanklichen) und der äußeren (realisierten) Sprachform festzustellen ist. Zwar sind die Mechanismen der stochastischen Wortvorhersage bis hin zur Präsentation der Wortfolgen als digitale Schriftzeichen technisch gleichermaßen auf die Modalitäten von Binärcode zurückzuführen. Weiterhin ausbleibend ist jedoch ein Prozessieren von Sprache »in Sprache« selbst, die nach Humboldt zur inneren wie äußeren Gesetzmäßigkeit des Denkens, Sprechens und Schreibens wird.

Obwohl die auch in der Informationstechnologie angewandte formalsprachliche Logik in Humboldts Theorie keine die Sprache begründende Stellung einnimmt, so ist sie doch ein inhärenter, strukturgebender Teil des Organismus Sprache. Explizit ist es die logische Grammatik, die er als eine der Antike entlehnte Theorie¹⁰ zur »Logisierung von Welt«¹¹ heranzieht. Sie dient einerseits der Formung des Gedanken und seiner Veräußerungen, indem sie eine syntaktische

9 Vgl. Ralph Krüger: Die Transformer-Architektur für Systeme zur neuronalen maschinellen Übersetzung. Eine popularisierende Darstellung. In: *trans-kom* 14/2 (2021), 282–319.

10 Vgl. Tilman Borsche: *Wilhelm von Humboldt*. München: Beck 1990: 144; Tilman Borsche: *Sprachansichten. Der Begriff der menschlichen Rede in der Sprachphilosophie Wilhelm von Humboldts*. Stuttgart: Klett-Cotta 1981, 209.

11 Siegfried J. Schmidt: *Sprache und Denken als Sprachphilosophisches Problem von Locke bis Wittgenstein*. Den Haag: Martinus Nijhoff 1968, 71.

Struktur sowie darauf aufbauende semantische Relationen von aufeinander bezogenen sprachlichen Einheiten ermöglicht.¹² Sie erfüllt jedoch vor allem deiktische Funktionen, da sie das individualisierte Subjekt mit anderen Subjekten oder Objekten verbindet. So tragen nach Humboldt etwa die Personalpronomina ›ich‹ und ›du‹ dazu bei, dass sich überhaupt ein Selbst- und Fremdbewusstsein herausbilden kann.¹³ Die Grammatik der Sprache dient Humboldt insofern als Vehikel der Erkenntnis und der Erschließung von Welt.

Hinsichtlich der holistischen Vorstellung der Sprache als zusammenhängendem Organismus muss jedoch unterstrichen werden, dass sämtliche Existenz in Sprache nur als *Energieia*, das heißt im Moment ihrer aktiven Hervorbringung zu begreifen ist, denn: »ihr ›Organismus‹ hat keine reale Existenz in der Natur, sondern nur eine ideale Existenz als ein Moment in der Aktualität (*Energieia*) der Rede.«¹⁴ Daraus folgt, dass die formalisierte Logik der Sprache bei Humboldt zwar zur Bildung des Bewusstseins und der sprachlichen Handlung beiträgt. Bedingt durch die untrennbare Verbindung der Logik zur Grammatik steht sie zugleich jedoch auch dem statischen Moment der Sprache als »todtes Machwerk wissenschaftlicher Zergliederung«¹⁵ nahe. Eben von jenem deskriptiv-statischen Verständnis der Sprache als einem »Vorath von Wörtern und ein System von Regeln«¹⁶ will sich Humboldt mit dem Programm der Sprache als Tätigkeit entschieden distanzieren.¹⁷ So betont er immer wieder ihr Moment der aktiven, subjektgebundenen

12 Vgl. Wilhelm Bondzio: Sprache als Arbeit des Geistes. In: Welke (Hg.): *Sprache, Bewußtsein, Tätigkeit*, 118; Renate Neurath: Grammatik als Verfahren. In: Welke (Hg.): *Sprache, Bewußtsein, Tätigkeit*, 134–136, 144–153; Borsche: *Sprachansichten*, 208–213; Borsche: *Wilhelm von Humboldt*, 144; Welke: *Sprache, Bewußtsein, Tätigkeit*, 31; Schmidt: *Sprache und Denken als Sprachphilosophisches Problem von Locke bis Wittgenstein*, 71f.

13 Vgl. Ferron: *Sprache ist Rede*, 110; Borsche: *Wilhelm von Humboldt*, 148f., 164–170. Borsche: *Sprachansichten*, 282–290; Menze: *Wilhelm von Humboldt*, 244–250.

14 Borsche: *Sprachansichten*, 220.

15 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 249.

16 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 249.

17 Vgl. Ferron: *Sprache ist Rede*, 122–124.

Hervorbringung, die sich von dem mechanischen Reduktionismus der traditionellen Sprachtheorie abgrenzt. Die Grammatik als logisches System verbleibt in seiner Theorie letztlich allein »im Gebiet der Möglichkeit«¹⁸, das heißt als historisch wachsendes, dokumentarisches Gebilde sprachlicher Konventionen, das erst im Moment des Sprechens als energetische Tätigkeit wieder zum Leben erweckt wird.

Die Integration der Maschine in den humboldtschen Sprachorganismus erscheint somit selbst hinsichtlich des logischen Teils der Grammatik als problematisch, da auch diese Teil eines Systems ist, das als Ganzes erst durch ihre aktive Hervorbringung zur Geltung kommen kann. Die maschinelle Erzeugung von Sprache ließe sich daher als ein *Ergon* bezeichnen, das zwar in der Lage ist, die Syntax und lexikalische Vielfalt der Sprache logisch stringent zu reproduzieren. Zugleich wird sie jedoch zum kalkülierten Werkzeug eines auf seine Phänomenologie ausgerichteten Imitationsspiels reduziert, das sich der anthropologischen Bedingung der in- und extrinsisch verankerten Sprache als allem Denken und Handeln vorausgehende Bedingung des Seins entledigt.

3. Weltlichkeit und kognitive Semantik

Im vorangegangenen Kapitel wurde bereits angeschnitten, dass die Sprache des Menschen bei Humboldt insbesondere der Erschließung von Welterfahrung dient. Im Folgenden soll die elementare Notwendigkeit von Welt im humboldtschen Kosmos genauer beleuchtet und dabei herausgestellt werden, welche grundlegenden Defizite computerbasierter Sprachmodelle auf den Mangel eben jener Weltlichkeit zurückzuführen sind.

Die Welt ist für Humboldt nicht nur die Stätte, an dem die durch die Grammatik organisierte Rede ihren Ausdruck finden kann. Sie ist vor allem der Ort, an dem das Subjekt eine spezifische Weltsicht beziehungsweise einen Standpunkt auf die Welt entwickelt. Da Humboldt

18 Horst Wessel: Ein Plädoyer für die Universalität der Logik. In: Welke (Hg.): *Sprache, Bewußtsein, Tätigkeit*, 96.

seine sprachanthropologische Theorie stets vom Individuum ausgehend konstruiert, ist diese Weltansicht auch die maßgebende Bedingung für die Entfaltung sowohl des menschlichen Charakters als auch die Erfüllung des humboldtschen Bildungsideals. Diesen voraus geht der Umstand, dass »aller objectiven Wahrnehmung unvermeidlich Subjectivität beigemischt ist«,¹⁹ die anschließend die fortlaufende semantische Orientierungsleistung des Verstehens in der zwischenmenschlichen Kommunikation erfordert.²⁰ Aufgrund der mangelnden Synchronität der vielfältigen Weltwahrnehmungen ist es nicht weiter verwunderlich, dass Humboldt den gesamten Organismus Sprache, ja überhaupt sämtliche Konstituierung der auf eine gemeinsame Objektivität abzielenden Weltwahrnehmung²¹ unmittelbar an die kommunikativen Funktionen der Sprache koppelt. Diese wiederum können nur in ihrem aktiven Vollzug, das heißt als Tätigkeit Ausdruck erhalten, insofern, als dass sich Subjekte mittels der Mechanismen der logischen Grammatik über die Eindrücke der Welt verständigen und schließlich deren repräsentativen Ausdrücke und Bedeutungen aushandeln. In jedem Fall wird die Weltlichkeit zur notwendigen Bedingung im Organismus Sprache, da nur sie den ›Stoff‹ der Referenzobjekte stellt, auf den in der Rede Bezug genommen wird. Denn der Gedanke, der sich beim Sprechen »aus der bewegten Masse des Vorstellens losreißt«²², wird erst in lautlicher Form als akustisches Phänomen objektiv für andere Subjekte wahrnehmbar.

Welches Verhältnis nehmen nun algorithmische Sprachmodelle gegenüber der weltlichen Instanz im humboldtschen Organismus ein? Und wie kommt eine Maschine zur Sprache, wenn sie selbst kein in der Welt tätiges Subjekt ist?

19 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 179.

20 Vgl. Menze: *Wilhelm von Humboldt*, 244–250; Borsche: *Wilhelm von Humboldt*, 167–170; Bondzio: *Sprache als Arbeit des Geistes*, 107–113; Schmidt: *Sprache und Denken als Sprachphilosophisches Problem von Locke bis Wittgenstein*, 75–77.

21 Das Aushandeln von Objektivität wird in der humboldtschen Kommunikationstheorie zum höchsten Erkenntnisziel erklärt. Vgl. Borsche: *Sprachansichten*, 277–279; Menze: *Wilhelm von Humboldt*, 241f, 257f.

22 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 26.

In der kognitionswissenschaftlichen Vision künstlich intelligenter Wesen gilt die Entwicklung sensorischer Kapazitäten als der Schlüssel zur Erschließung von Weltlichkeit, die es Maschinen erlauben, Welt nicht nur zu ertasten, sondern diese auch kausal zu verstehen. Insbesondere in der die Kybernetik begleitenden Embodiment-Theorie gilt die Anbindung der Maschine an die Welt als die Einsicht, »[d]ass unsere Körper, Gehirne und Interaktionen mit unserer Umwelt die meist unbewusste Grundlage für unsere Alltagsmetaphysik bilden [...]«. ²³ Ansätze einer solchen Implementierung finden inzwischen auch bei sprachverarbeitenden Systemen Anwendung. ²⁴ In Analogie zu Humboldts zirkulärem Organismus scheint die Vernetzung von Sprache, Körper und Welt zum Hoffnungsträger entscheidender Fortschritte in der Entwicklung künstlicher Intelligenzen zu werden. In bemerkenswerter Übereinstimmung zu Humboldts Organismus-Theorie wird die Sprache schrittweise zur zentralen Instanz eines neuen maschinellen Denkens und nun auch körperlichen Handelns erklärt.

Ob Ansätze dieser Theorie jemals die Hürde wahrer Intelligenz überwinden können, die mehr als eine behaviouristische Kopie menschlicher Fähigkeiten zu bieten hat, bleibt fraglich. Weiterhin schnell durchschaubar bleiben jedenfalls Systeme wie GPT-4, die zwar zunehmend multimodale Qualitäten annehmen, ²⁵ sich in ihrer Architektur jedoch weiterhin als körperlose Sprachmodelle verstehen. Bedingt durch den

23 George Lakoff/Mark Johnson: *Philosophy in the Flesh. The Embodied Mind and Its Challenge to Western Thought*. New York: Basic Books 1999, 17. Übers. d. Verf.

24 Das Projekt »PaLM-SayCan« des Google Research Labs zum Beispiel verbindet klassische Ansätze der Kybernetik mit einem *Large Language Model*, dessen sprachliche Kompetenz als Schnittstelle zur Welt dienen soll. Mittels sensorischer Integration in die physische Welt soll ein Roboter lernen, die Defizite der lediglich syntaktischen Symbolverarbeitung zu überwinden und ein semantisches Verständnis der Realität und ihrer sprachlichen Repräsentation zu entwickeln. Vgl. Michael Ahn et al.: *Do As I Can, Not As I Say. Grounding Language in Robotic Affordances* (2022). <https://doi.org/10.48550/arXiv.2204.01691> (letzter Zugriff am 12.08.2023)

25 Dazu zählen Schnittstellen der Bild-, Video- und Tonverarbeitung sowie die Anbindung an diverse Onlinedienste und Wissensdatenbanken.

Entzug eben jener Weltlichkeit gerät die Vorstellung eines maschinellen Denkens in Sprache, das in enger Verbindung zur Welt und ihrer agentien Handlungsmöglichkeiten steht, nach Humboldts Vorstellung bereits unmittelbar ins Wanken. Trotz dieser sprachtheoretischen Einschränkung wird der Aufschwung großer Sprachmodelle mittlerweile durch ein breites Repertoire phänomenologisch ausgerichteter Studien begleitet, die – wenig überraschend – nicht nach den ontologischen Voraussetzungen des Geistes, sondern nach den pragmatisch vergleichbaren Fähigkeiten von künstlichen Sprachsystemen fragen. Eine vom Microsoft-Research-Team durchgeführte Studie etwa attestiert dem Sprachmodell GPT-4 erste »Funken von künstlicher Intelligenz«²⁶, da es bereits eine ganze Reihe an sprachlich messbaren Kompetenzen beherrsche.²⁷ Besonders hervorzuheben ist hierbei die These, dass das Sprachmodell trotz des offensichtlichen Mangels an Verkörperung dazu in der Lage sei, Aufgaben auszuführen, die ein Verständnis von Welt erfordern.²⁸ Gemeint sind Interaktionen mit der Welt, die über die Sprache als Interface vollzogen werden. In einem Experiment gelingt es GPT-4 beispielsweise, mittels sprachlicher Deskriptionen den Weg durch ein virtuelles Areal zu finden und anschließend eine korrekt angeordnete Karte mit allen passiertten Räumen zu zeichnen.²⁹ Dem Sprachmodell selbst steht hierbei lediglich das Feedback des Menschen zur Verfügung, der angibt, wo sich GPT-4 zum jeweiligen Zeitpunkt befindet.³⁰ Das Beispiel soll das räumliche Vorstellungsvermögen des Algorithmus illustrieren, das über das logische Prozessieren natürlicher Sprachsequenzen eigenständig erschlossen wird. Es illustriert jedoch vor allem ein Phänomen, das der Kognitionsforscher Stevan Harnad 1990 als das ›Symbol Grounding Problem‹ beschreibt. In seinem Aufsatz

26 Vgl. Bubeck et al.: Sparks of Artificial General Intelligence. Early experiments with GPT-4 (2023). <https://doi.org/10.48550/arXiv.2303.12712> (letzter Zugriff am 28.06.2023). Übers. d. Verf.

27 Vgl. Bubeck et al.: Sparks of Artificial General Intelligence.

28 Vgl. Bubeck et al.: Sparks of Artificial General Intelligence, 49, 53.

29 Vgl. Bubeck et al.: Sparks of Artificial General Intelligence, 51f.

30 Vgl. Bubeck et al.: Sparks of Artificial General Intelligence, 51f.

führt er den Mangel an reflektiertem, semantischem Denken der Rechenmaschinen auf die Funktionsweise symbolischer Systeme zurück, die nicht von der Bedeutung sprachlicher Einheiten, sondern von deren regelbasierten Zusammensetzung ausgehen.³¹ Seine These lautet: »Um geerdet zu sein, muss das Symbolsystem um nicht-symbolische, sensomotorische Kapazitäten erweitert werden [...].«³² Ein komplexes, sprachliches Modell der Welt, das nicht nur mit symbolischen Repräsentationen arbeitet, sondern auch ihre Semantik beziehungsweise Ideen verstehen kann, ist demnach nur durch eine kognitive Verbindung zur Wirklichkeit realisierbar. Seine Theorie entwickelt Harnad unter anderem anhand John R. Searles ›Chinese Room Argument‹, das sich als Kontrastprogramm zum behavioristischen Turing-Test versteht. Searle bemängelt insbesondere das ausbleibende Kriterium des semantischen Verstehens und macht auf die rein symbolisch-syntaktische Natur des maschinellen Denkens aufmerksam.³³ Er postuliert, dass eine regelbasierte Symbolverarbeitung für sich genommen keine Bedeutung herstellen könne, weil ihr die Intentionalität und Kausalität des Denkens fehle.³⁴ Eine ähnliche Haltung vertritt auch Noam Chomsky, der angesichts der jüngsten Errungenschaften von großen Sprachmodellen das weiterhin offensichtliche Ausbleiben eines kausal-deterministischen Denkens kritisiert.³⁵ Ein Lernen, das lediglich die Deskription und Vorhersage von Ereignissen beherrsche, könne we-

31 Vgl. Harnad, Stevan: The Symbol Grounding Problem (1999). <https://doi.org/10.48550/arXiv.cs/9906002> (zuletzt abgerufen am 12.08.2023).

32 Harnad, Stevan: Symbol grounding problem [sic!] (2007). <http://dx.doi.org/10.4249/scholarpedia.2373> (zuletzt abgerufen am 12.08.2023).

33 Vgl. John R. Searle: Minds, brains, and programs. In: *Behavioral and Brain Sciences* 3/3 (1980), 1–19.

34 Vgl. Searle: Minds, brains, and programs, 1–19.

35 Vgl. Machine Learning Street Talk: Noam Chomsky, The Ghost In The Machine (2023). <https://www.youtube.com/watch?v=axuGfh4UR9Q/https://whimsical.com/mlst-chomsky-transcript-WgFJLguL7JhzyNhsgdwATy> (zuletzt abgerufen am 28.06.2023).

der kausale Zusammenhänge, noch physikalische Gesetzmäßigkeiten verstehen.³⁶

Die Symbolisierung der Sprache spielt auch in der humboldtschen Theorie eine Rolle, insofern, als dass sie die Verknüpfung von Gedanken und Welt in der Rede realisiert.³⁷ Gegenüber der Trivialisierung der Sprache als ein fixiertes, grammatikalisches Gebilde sieht er jedoch auch die Gefahr, »die Wörter als blosse Zeichen anzusehen«³⁸. Er behandelt dies als den »Grundirrtum, der alle Sprachwissenschaft und alle richtige Würdigung der Sprache zerstört.«³⁹ Entscheidend ist für Humboldt auch hier die genetische Verknüpfung der äußeren und inneren Sprachform. Als äußere Form bezeichnet er das gesprochene Wort, dem als arbiträres Zeichen eine vermittelnde Funktion in der stofflichen Welt zukommt.⁴⁰ Das Wort als Symbol hingegen ist eine mentale Idee, die die subjektive Semantik des Denkens repräsentiert.⁴¹ Die Konstitution von Bedeutung ist auch in diesem Fall nur durch das enge Zusammenspiel von Denken (als Symbol) und Sprechen (als Zeichen) möglich. Beide bedingen einander, da das formale, äußere Zeichen auf die semantische Weltvorstellung in Gedanken verweist, genauso, wie der Gedanke erst durch ein objektives Zeichen kommuniziert und im historischen, soziokulturellen Prozess in der Gesellschaft semantisch differenziert werden kann. Die Semantik von an die Welt gebundenen Eindrücken beziehungsweise deren Begriffen lässt sich in dieser Konsequenz auch »niemals ursprünglich im Lexikon nachschlagen«⁴², da Bedeutung nur im energetischen Prozess von Denken und Sprechen als subjektgebundene, kommunikative Tätigkeit vorstellbar ist. Das Symbol dient gleichermaßen wie die logische Grammatik lediglich als

36 Vgl. Machine Learning Street Talk: Noam Chomsky, *The Ghost In The Machine* (2023).

37 Vgl. Borsche: *Sprachansichten*, 258–260.

38 Humboldt: *Grundzüge des allgemeinen Sprachtypus*, 428.

39 Humboldt: *Grundzüge des allgemeinen Sprachtypus*, 428.

40 Vgl. Humboldt: *Grundzüge des allgemeinen Sprachtypus*, 428.

41 Vgl. Humboldt: *Grundzüge des allgemeinen Sprachtypus*, 428.

42 Borsche: *Wilhelm von Humboldt*, 153.

Teil eines zusammenhängenden Organismus, in dem »Alles durch Jedes und Jedes durch Alles bestimmt wird.«⁴³

In ihrer Argumentation haben die Ansätze Harnads, Searles, Chomskys und Humboldts zweierlei gemeinsam. Einerseits werden die Konzepte eines komplexen Denkens an die Fähigkeit der kognitiven Wahrnehmung gebunden, die konsequenterweise nur in der Wirklichkeit verortet werden kann. Andererseits verweisen sie auf das Defizit der maschinellen Symbolverarbeitung, das sich der Notwendigkeit einer auf weltliche Erfahrung referierenden Semantik entzieht. In Folge dieser Einschränkungen lässt sich nun auch das zuvor erwähnte Beispiel von GPT-4 relativieren, bei dem das Sprachmodell ein virtuelles Areal erkunden und rekonstruieren soll. Zwar beherrscht das Programm die Fähigkeit, den Anweisungen zu folgen und die geometrische Raumanordnung nicht nur zu erkunden, sondern diese auch korrekt zu beschreiben. Es bleibt jedoch gefangen im selbstbezüglichen »Symbolkarussell«⁴⁴, weil es diese Erkenntnisse lediglich aus logischen, probabilistischen Zeichenrelationen und nicht kognitiv aus der Welt beziehen kann.⁴⁵ Perzeptiv betrachtet bleibt die Maschine somit hinter dem Menschen zurück, weil sich das kausale Analogon der Wirklichkeit scheinbar nicht unbeschadet in die binäre Logik übersetzen lässt. Auch wird eine Verbindung zur Welt erst durch den Menschen als Schnittstelle hergestellt. Die Maschine ist somit existenziell determiniert vom Menschen, der als Programmierer und Initiator des Codes außerdem die widerspruchslöse Deutungshoheit über Syntax und Semantik trägt.

43 Humboldt: Grundzüge des allgemeinen Sprachtypus, 394.

44 Hannes Bajohr: Dumme Bedeutung. Künstliche Intelligenz und artifizielle Semantik. In: *Merkur* 76/882 (2022), 71.

45 So schränken auch Bubeck et al. im Kontext komplexer, mehrschrittiger Aufgabenstellungen ein: »Das Modell stützt sich auf einen lokalen, effizienzorientierten Prozess der Generierung des nächsten Wortes, ohne ein globales oder tiefes Verständnis der Aufgabe oder der Ausgabe.« (Bubeck et al.: *Sparks of Artificial Intelligence*, 80. Übers. d. Verf.).

4. Generative Individuation

Der Ausschluss von der weltlichen Sphäre hat nicht nur den Verlust des semantischen Verständnisses zur Folge. Er bedeutet konsequenterweise auch den Verlust sämtlicher Individuation und charakterlichen Bildung, die nämlich erst durch spezifische Weltansichten möglich werden. Obwohl Sprachmodelle wie GPT-4 dazu in der Lage sind, persönliches Einfühlungsvermögen zu zeigen oder auf Anweisung unterschiedliche soziale Rollen einzunehmen, so scheint es doch offensichtlich, dass sie kein Ergebnis unmittelbar kognitiver Weltwahrnehmung sein können. Ihre Fähigkeit der persönlichen Interaktion entspringt keinem unmittelbaren Subjekt, sondern schlicht den Daten, mit denen sie trainiert wurden.⁴⁶ Sie sind somit eine »imitatio naturae«⁴⁷, die die bloße »Wiederholung des Realen«⁴⁸ als eine stetige Rekombination von Zeichenfolgen im trainierten Datensatz, die das Natürliche phänomenologisch anstrebt, dieses existenziell jedoch nie erreicht. Das Hervorbringen von Sprache und Individuation kann in dieser Konsequenz allerhöchstens als Simulation zum Vorschein kommen, die vom *Ergon* Sprache als diffuse Datenmenge ohne kognitiven Kontext erzeugt wird. Teil dieser Trainingsdaten sind unter anderem das ›Ich‹ und das ›Du‹, die nach Humboldt nur der unmittelbaren Erkenntnis des Selbst und des Gegenübers entspringen können.⁴⁹ Generalisierte Sprachmodelle wie ChatGPT vermögen diese Verhaltensmuster

46 Saketh Reddy Karra/Son The Nguyen/Theja Tulabandhula: Estimating the Personality of White-Box Language Models (2023). <https://doi.org/10.48550/arXiv.2204.12000> (letzter Zugriff am 02.08.2023), 24: »Unsere Experimente haben gezeigt, dass die Sprachmodelle unterschiedliche Persönlichkeitsmerkmale aufweisen, die die für ihr Training verwendeten Datensätze widerspiegeln.« (Übers. d. Verf.).

47 Vgl. Hannes Bajohr: *Schreibenlassen. Texte zur Literatur im Digitalen*. Berlin: August 2022, 156.

48 Vgl. Bajohr: *Schreibenlassen*, 156.

49 Vgl. Menze: *Wilhelm von Humboldt*, 244–250; Ferron: *Sprache ist Rede*, 110, Borsche: *Wilhelm von Humboldt*, 148f., 164–170. Borsche: *Sprachansichten*, 282–290.

personalisierter Sprache und ihrer sozialen Rollen zwar zu adaptieren, bleiben in ihrem ›Wissen‹ in Form der Möglichkeit potenzieller Sprachaufgaben letztlich jedoch enzyklopädisch. Die Situiertheit der Sprache sowie aller sich in dieser abzeichnenden Individualität wird reduziert auf das Alltagsdeutsch als den lexikalischen »Vorrath von Wörtern«⁵⁰ und deren regelbasierten Zusammenhänge. Mit dieser durch Transformer-Technologien einhergehenden Anonymisierung und Dekontextualisierung individualsprachlicher Texte scheint sich schließlich ein technologischer Blick auf Sprache durchzusetzen, der an den informationstheoretischen Antisubjektivismus Max Benses erinnert.⁵¹ An Stelle des abstrakten, deutungsbedürftigen Ausdrucks rückt nämlich die Rationalität berechenbarer Sprachsysteme, die ohne »subjektive Irrtümer«⁵² und Verzerrungen der individuellen Weltanschauung ›funktionieren‹ sollen. Ganz im Stile vermeintlich ›neutraler‹ Suchalgorithmen gilt es eine spezifische Individualität hier nicht zu erzeugen, sondern vielmehr zu vermeiden.

Als Gegenentwurf zum depersonalisierten Modell enzyklopädisch trainierter Sprachsysteme stehen zunehmend auch solche Dienste zur Verfügung, die mittels selektiven Trainings die Illusion realer Gesprächspartner:innen erzeugen möchten.⁵³ Auch solche Ansätze müssen jedoch als ›stochastische Papageien«⁵⁴ einer sequenziellen Wortvorhersage behandelt werden, die lediglich Simulationen von Individuen zu erzeugen vermag. Nach Humboldt dürfte auch in diesem

50 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 180.

51 Vgl. Philipp Schönthaler: *Die Automatisierung des Schreibens & Gegenprogramme der Literatur*. Berlin: Matthes & Seitz 2022. 165f., 179.

52 Schönthaler: *Die Automatisierung des Schreibens & Gegenprogramme der Literatur*, 166.

53 Webdienste wie »character.ai« (<https://beta.character.ai/> [zuletzt abgerufen am 09.11.2023]) etwa wollen authentische Schriftwechsel mit Charakteren wie beispielsweise William Shakespeare, Albert Einstein oder Elon Musk ermöglichen.

54 Vgl. Emily M. Bender et al.: On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? (2021). <https://doi.org/10.1145/3442188.3445922> (zuletzt abgerufen am 12.06.2023).

Kontext gelten, dass die »Verschiedenheit der Köpfe«⁵⁵ und die leibhaftige Identität des Individuums in seiner mannigfaltigen Komplexität wissenschaftlich kaum gefasst und gebändigt werden können.⁵⁶ Der Organismus Sprache als holistisches Gebilde ist auch an dieser Stelle nicht hintergebar, eine Deskription seiner einzelnen Teile allenfalls ein Artefakt. So folgert auch der Sprachwissenschaftler Jürgen Trabant: »Die erschöpfende Beschreibung eines Individuums ist unmöglich: »individuum est ineffabile.«⁵⁷ Der behavioristische Ansatz der Computerwissenschaften, der das menschliche Verhalten beschreiben, modellieren und anschließend auch maschinell reproduzierbar machen möchte, ist somit nicht einzulösen, da sich der Mensch als lediglich verhaltenspsychologisch betrachtetes *Ergon* nicht begreifen lässt.⁵⁸

5. Das Ergon Schrift

Die bisherige Untersuchung hat gezeigt, dass die Sprache als Tätigkeit bei Humboldt insbesondere vom mündlichen Sprechakt ausgeht. Im Folgenden sollen die konkreten medialen Bedingungen für den Sprachorganismus näher betrachtet werden, um zu klären, ob der Vollzug von Sprache als *Energeia* überhaupt über Schriftlichkeit möglich ist.

-
- 55 Wilhelm von Humboldt: Theorie der Bildung des Menschen. Bruchstück. In: Albert Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 1. Berlin: Behrs 1903, 287.
- 56 Vgl. Wilhelm von Humboldt: Über den Nationalcharakter der Sprachen. Bruchstück. In: Albert Leitzmann (Hg.): *Wilhelm von Humboldts gesammelte Schriften*, Bd. 4. Berlin: Behrs 1905, 421: »Die Untersuchung dieser Individualitaet, ja sogar ihre genauere Bestimmung in einem gegebenen Falle ist das schwierigste Geschäft der Sprachforschung. Es ist unläugbar, dass dieselbe, bis auf einen gewissen Grad, nur empfunden, nicht dargestellt werden kann [...]«.«
- 57 Jürgen Trabant: *Apeliotes oder der Sinn der Sprache. Wilhelm von Humboldts Sprach-Bild*. München: Fink 1986: 195.
- 58 Vgl. Schönthaler: *Die Automatisierung des Schreibens & Gegenprogramme der Literatur*, 165f.

Wenn Humboldt von der Schrift spricht, so definiert er diese gegenüber der gesprochenen Sprache in erster Linie als ein System von Zeichen, »welche bestimmte Wörter in bestimmter Folge andeuten.«⁵⁹ Als semiotisches System dient es der Darstellung von Begriffen beziehungsweise deren Lautform und kann somit auch als Ideenbeziehungsweise Lautschrift bezeichnet werden.⁶⁰ Eine besondere Bedeutung kommt dem Medium Schrift bei Humboldt jedoch im Kontext der vergleichenden Sprachstudien zu, in denen er ihren Wert als wissenschaftliches Dokumentationsmedium unterstreicht.⁶¹ Gegenüber der stetigen Mouvance und Unschärfe mündlicher Überlieferungen fungiert sie als überzeitlicher Bedeutungsträger, der die diachrone Erforschung sprachlicher Kulturen im weiteren Umfang überhaupt erst möglich macht.⁶² Neben ihren deskriptiven Funktionen zählt die Schrift als Gegenspieler zur gesprochenen Rede jedoch auch als notwendige Bedingung zur Entwicklung und Ausdifferenzierung der nationalen Sprachen. Ihren Grad der Vervollkommnung können sie nämlich erst durch das wechselseitige Zusammenspiel von Laut und Schrift erreichen.⁶³ Die Schrift führt nämlich nicht nur »der Seele die Articulation der Töne vor, indem sie die articulirten Töne vereinzelt und bezeichnet.«⁶⁴ Sie ermöglicht auch eine »neue Einsicht in die Natur der Sprache«⁶⁵, da sie die Worte gegenüber dem vergänglichen Schall der Artikulation »klarer und anschaulicher«⁶⁶ fixiert. Sie dient somit nicht

59 Wilhelm von Humboldt: Über den Zusammenhang der Schrift mit der Sprache. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 5, 34.

60 Vgl. Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 39.

61 Vgl. Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 35.

62 Vgl. Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 35.

63 Vgl. Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 35–39.

64 Wilhelm von Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 5, 115.

65 Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 116.

66 Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 115.

nur als ein die Sprache dokumentierendes, sondern auch als ein diese formendes Reflexionsmedium, das neue Qualitäten der Beschreibung und des Verständnisses von Sprache ermöglicht.

Neben dem artikulierten Wort trägt nun also auch das Medium der Schrift einen bedeutenden Anteil an der Sprachtheorie Humboldts. Trotz der wechselseitigen Synergieeffekte von Schrift und Artikulation für die Ausgestaltung der Sprache kommen ihnen im Organismus der Sprache jedoch mitunter völlig verschiedene Funktionen zu. Während das »tönende Wort«⁶⁷ etwa die unmittelbare »Verkörperung der Gedanken«⁶⁸ vollzieht und dadurch zum Vehikel der energetischen Sprachtätigkeit wird, erfüllt die Schrift insbesondere deskriptive Funktionen, die der Erhaltung der lexikalisch-grammatischen Form der Sprache und ihrer Vielfalt dienen. Als physikalisch überzeitliches Medium vermag die Schrift die Sprache und ihre Werke zwar besonders gut zu dokumentieren. Gegenüber dem Charakter des Lauts, der die Spontanität und Vergänglichkeit der Gedanken »in unmittelbarer Lebendigkeit«⁶⁹ widerspiegelt, steht sie jedoch auch dem Werk der Sprache als »mumi-entartige Aufbewahrung«⁷⁰ nahe. Sie erscheint insofern als ein *Ergon*, weil sie in ihrer Statik nicht nur der Natur des sich ständig erneuernden Denkens widerspricht, sondern auch keine unmittelbare, organische Verbindung mit der inneren Denkform des Menschen eingeht. Zwischen Gedanken und Schrift tritt bei Humboldt stets das gesprochene Wort, denn die Schrift »ist doch immer nur Bezeichnung des schon durch die Sprache geformten Gedanken.«⁷¹ Schrift wird hier also auf-

67 Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 109.

68 Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 109.

69 Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 34.

70 Humboldt: Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 7. Berlin: Behrs 1907, 41.

71 Humboldt: Über den Zusammenhang der Schrift mit der Sprache, 35.

gefasst als das Produkt einer Übersetzung⁷² und kann insofern nicht als unmittelbare Form des energetischen Denkens herangezogen werden.⁷³ In der Hierarchie der Sprache erscheint das Erzeugnis Schrift demzufolge als ein dem Sprechen nachgelagertes Medium, das nicht dem Gedanken, sondern dem Sprechen entspringt. Auch die Vollendung der Sprache vermag – trotz der historischen Leistungen der Schrift – letztlich also nur das Moment ihrer aktiven Hervorbringung in Form der Artikulation zu leisten. Das Sprechen dient als »Schlussstein«⁷⁴ des energetischen Sprachorganismus, den Humboldt gegenüber den konventionellen Ansätzen der Grammatiktheorie perspektivisch nur als Tätigkeit verstanden wissen möchte.⁷⁵ Mit dieser Festlegung als Tätigkeit unterstreicht er zugleich, dass die Sprache nur vom sprechenden Individuum und dessen Subjektivität (Weltsicht) aus gedacht werden kann.⁷⁶

Wie lassen sich diese Erkenntnisse nun auf den Untersuchungsgegenstand der schreibenden Maschinen anwenden? Nachdem festgestellt wurde, dass Algorithmen über keinen inneren, semantischen Sprachsinn verfügen, so kann nun auch ihr äußeres Sprachvermögen auf einen Mangel sprachanthropologischer Grundvoraussetzungen zurückgeführt werden. Wenn Sprachmodelle wie GPT-4 lediglich auf Schrift zurückgreifen, um wiederum nur Schrift zu erzeugen, so operieren sie auf der unteren Ebene der sprachtheoretischen Hierarchie, die Sprache lediglich als strukturalistisches Erzeugnis auffasst. Auch an diesem

72 Vgl. Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 112.

73 Humboldt postuliert: »Allein das tönende Wort ist gleichsam eine Verkörperung des Gedanken, die Schrift eine des Tons.« (Humboldt: Über die Buchstabenschrift und ihren Zusammenhang mit dem Sprachbau, 109).

74 Humboldt: Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts, 152.

75 Vgl. Tintemann/Trabant: *Wilhelm von Humboldt*, 87; Neurath: Grammatik als Verfahren, 144–153.

76 Vgl. Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 249; Humboldt: Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts, 46.

Beispiel zeigt sich der phänomenologische Reduktionismus generierter Sprache, der den intrinsischen Prozess des Denkens sowie den subjektiven Akt des Sprechens völlig ausklammert. Die auf diese Weise um ihren Organismus beraubte Sprache wird zum messbaren Gegenstand einer probabilistischen Datenverarbeitung erklärt. Sie ist damit schließlich das künstliche Produkt einer technischen Verwissenschaftlichung der Sprache, denn: »Wenn sie sich von der Individualität des einzelnen Sprechaktes entfernt, wird sie zum menschlichen Konstrukt.«⁷⁷ Das Maschinenschreiben ist in dieser Konsequenz nicht als *Energeia* vertretbar, da es mit dem interdependenten Organismus der Sprache bricht und lediglich sein Produkt der Schriftsprache herausgreift. Sie ist eine Sprache ohne Sprecher:in.

6. Maschinenschreiben als *Energeia* und die Prozessualität der Schrift

Im bisherigen Verlauf dieses Beitrags wurde das maschinelle Schreiben aus dem strengen Blickwinkel einer anthropologischen Sprachtheorie betrachtet. Dabei konnte herausgestellt werden, dass Computer den sowohl geistigen als auch organisch-kausalen Bedingungen der Sprache nicht entsprechen können. Die Möglichkeit energetischer Maschinensprache kann hierbei nämlich weder aus der logischen Symbolsprache noch aus der Analyse empirischer Daten oder einer probabilistischen Simulation heraus gerechtfertigt werden. Offen bleibt jedoch, was das eigentliche Wesen der Sprache, des Geistes und nicht zuletzt das spezifisch Menschliche ausmacht. Indem Humboldt eine relationale Bestimmung der dialektischen Einheiten seines Sprachorganismus vornimmt, umgeht er nämlich die Notwendigkeit, etwa das Wesen der Sprache an sich sowie ihren Ursprung zu bestimmen. Die Sprache wird folglich zu einer »dem Menschen auf unerklärliche Weise gegebene Wesensqualität.«⁷⁸ Die Frage nach ihrem Ursprung »bleibt in sich selbst unerklär-

77 Ferron: *Sprache ist Rede*, 19.

78 Welke: *Sprache als Arbeit des Geistes*, 48.

bar«⁷⁹ sowie »notgedrungen inhaltsleer, da sie es nicht schafft, die Sprache als Forschungsgegenstand restlos aus sich selbst heraus zu bestimmen.«⁸⁰ In gleicher Weise gilt für Humboldt auch die Geisteskraft des Menschen als apriorisch gesetzte, wissenschaftlich nicht hintergehbare Bedingung.⁸¹

Die Übertragung einer solchen anthropozentrischen Sprachtheorie auf die Maschine erweist sich nun gleich in doppelter als problematisch. Einerseits ist der Computer offensichtlich weder menschähnliches Subjekt, noch kann er ein einschlägiges, geistestheoretisch fundiertes Äquivalent des Denkens vorweisen, das nicht auf Einsen und Nullen beruht. Andererseits ist nicht einmal ersichtlich, wie sich der menschliche Geist und seine Veranlagung zum Sprechen überhaupt distinkt feststellen und anschließend vergleichen lassen soll. Bemerkenswert erscheint der Umstand, dass auch die vermeintliche Rationalisierung des Geistes und seiner Sprache durch die Erforschung künstlicher neuronaler Netzwerke bisher nur wenig zu dieser Entschlüsselung beitragen konnte. Womöglich zeugt auch die naturwissenschaftliche Benennung des *Black-Box*-Phänomens, welche die kaum noch greifbare Komplexität der tief geschichteten Neuronenverbindungen beschreibt,⁸² von der Unhintergebarkeit des menschlichen Geistes und seiner Sprache.

Ausgehend von diesen Einschränkungen soll im Folgenden versucht werden, die Theorie Humboldts im Zeitalter anbrechender künstlicher Intelligenz neu zu deuten, indem vor allem der kommunikative Vollzug der Sprache als *Energieia* in den Mittelpunkt gestellt wird. Angesichts der offensichtlichen paradigmatischen Verschiebungen durch die maschinelle Erschließung der Sprache müssen hierzu zwar die Vorbehalte der Phänomenologie und klassischen Geistestheorie zurückgestellt werden. Im Gegenzug bietet sich jedoch die Möglichkeit, die maschinelle Hervorbringung der Sprache aus dem Blickwinkel einer neuen Agentialität

79 Ferron: *Sprache ist Rede*, 143.

80 Ferron: *Sprache ist Rede*, 26.

81 Vgl. Menze: *Wilhelm von Humboldt*, 97.

82 Vgl. Davide Castelvecchi: The Black Box of AI. In: *Nature* 538/7623 (2016), 20–23.

zu betrachten. Zudem können Sprachmodelle im interaktiven »Handlungsgefüge zwischen Mensch und Maschine«⁸³ verortet und auf ihre sprachtheoretischen Kompetenzen hin befragt werden.

Der Sprachorganismus bei Humboldt ist ein System von sich wechselseitig bedingenden Einheiten. Teil dieses Systems ist auch die Schrift, die dem Erhalt von Wortschatz und Grammatik sowie der diachronen Dokumentation und Reflexion über Sprache dient. Als ein die historische Fixierung anstrebendes Medium steht es jedoch dem energetischen Sprechen gegenüber, das sich durch seine stetige Erneuerung und enge Bindung zur Gedankenform auszeichnet. Eine Übertragung dieses Paradigmas auf schreibende Maschinen erscheint zunächst fraglich, weil das Schrift-Sein unmittelbar auf das sprachliche *Ergon* verweist. Zugleich zeichnen sich Sprachmodelle wie GPT-4 jedoch dadurch aus, dass ihre Schrift weder die reine Übersetzungsleistung eines originellen Sprechakts darstellt, noch als unmittelbares Abbild bereits geschriebener, historischer Werke gilt. Vielmehr zeugt die generative Sprache einschlägiger Transformer-Modelle von einer fortlaufenden Aneinanderreihung von Wörtern im grafischen Interface, deren Kombination iterativ einmalig ist. Ihre sprachliche Einmaligkeit schöpft sich aus der Tatsache, dass neuronale Netzwerke eben keinen mittelbaren Datenbankgleich vornehmen, wie es etwa noch Weizenbaums, eingangs erwähntes ELIZA-Modell praktizierte. Stattdessen werden Aussagen durch ein komplexes Netzwerk von untereinander »konkurrierenden« Approximationsmechanismen ständig neu generiert. Zugleich könnte auch die sichtbare Prozessualität der GPT-Modelle als Indiz dafür betrachtet werden, dass das sequenzielle Schreiben der Maschine eine Form von Bewegung beherrscht, die nicht nur dem natürlichen Denken in Sprache ähnelt,⁸⁴ sondern auch der humboldtschen Idee der sich stetig neu schöpfenden Tätigkeit nahesteht. Die Tätigkeit in Sprache

83 Wiebke Vorrath: Unter der Oberfläche? Programmierte Schriftlichkeit in digitaler Lyrik. In: Bartelmus/Nebbrig (Hg.): *Schriftlichkeit*, 58.

84 Auch das menschliche Denken in Sprache erfolgt Wort für Wort, da es die Vorstufe der ebenfalls sequenzielle erfolgenden Rede bildet. Vgl. Bajohr: *Schreiben lassen*, 158, 163, 197.

beschreibt Humboldt als »ein fortschreitendes Entwickeln, eine blossere innere Bewegung, in der nichts Bleibendes, Stätiges, Ruhendes angenommen werden kann.«⁸⁵ Eben jene Bewegtheit ist nicht weniger als die treibende ›Kraft‹ seiner Sprachtheorie, vor der sich das *Ergon* als das die historische Fixierung anstrebende Werk der Sprache abgrenzt. Aufgabe der geistigen Bewegung beziehungsweise der *Energeia* ist es nun, dem »totden Gerippe«⁸⁶ des lexikalischen Wortschatzes einer Sprache weltliche Geltung zu verschaffen, indem es in das Moment eines Gedankengangs beziehungsweise der sprachlich realisierten Rede versetzt wird.⁸⁷ Im Fall von GPT-Modellen ist es das geschriebene Wort, in dem sich eine solche Form der Sprachtätigkeit als prozessierte Sprachtätigkeit offenbart. Das in die gewichteten Verknüpfungen des neuronalen Netzwerks codierte *Ergon* Sprache – als Speicher von Lexikon, Grammatik und Syntax – wird in das aktive Moment geschriebener Sprache versetzt. Oder organisatorisch gesprochen: Im Übergang vom Toten zum Lebendigen offenbart sich der Akt des Schreibens. Es ist jedoch ein Schreiben ohne Subjekt, ohne Autor:in, ohne intuitiver Intention und kognitiver Überschneidung zur tastbaren Wirklichkeit. Das maschinelle Schreiben birgt somit »etwas Prozesshaftes, das sich im Schreibakt selbst ausdrückt, aber eben auch ohne Subjekt dem Schriftlichen zukommt.«⁸⁸ Es ist kein mittelbares Subjekt, aber auch kein gänzlich determinierbares Werkzeug, da es nicht explizit programmiert werden kann.⁸⁹ Man könnte es folglich bezeichnen als die subjektlose Schnittmenge zwischen *Ergon* und *Energeia*, als das zugleich energetische aber auch nicht-energetische Zusammenwirken von Struktur (*Ergon*) und Realisation (*Energeia*) der Sprache. Eine strenge genetische Trennung des humboldtschen Modells scheint für das maschinelle Schreiben somit nicht mehr möglich.

85 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 154.

86 Humboldt: Über die Verschiedenheiten des menschlichen Sprachbaues, 147.

87 Vgl. Borsche: *Sprachansichten*, 220.

88 Bartelmus/Nebbrig: Schriftlichkeit, 12.

89 Vgl. Bajohr: *Schreibenlassen*, 153.

7. Kommunikative Subjektivation

Nach der Untersuchung der Medialität der Schrift sowie deren Verhältnis zur *Energeia* soll die Möglichkeit energetischen Schreibens im Folgenden auch aus einer kommunikationssoziologischen Perspektive neu bewertet werden. Die folgende Fragestellung lautet, auf welche Weise dem maschinellen Schreiben eine Form virtueller Agentialität zugesprochen werden kann, die an Stelle des sprechenden Subjekts im Sprachorganismus tritt. Die Untersuchung der Sprachtheorie Humboldts hat gezeigt, dass die Wechselseitigkeit der Rede als Grundvoraussetzung für die Organisation der Welt in benennbare beziehungsweise adressierbare Objekte und Subjekte dient.⁹⁰ Ausgehend von dem menschlichen Bestreben nach Verständigung⁹¹ stabilisiert sie die Bedeutung der Dinge, indem sie die Subjekte gedanklich synchronisiert und das Verständnis sicherstellt.⁹² Ermöglicht wird dies jedoch nur durch das vorausgehende Erkennen der Subjekte untereinander, die sich als sprechende Akteure mittels der grammatischen Personalpronomina⁹³ über die Welt der Objekte erheben. Denn erst »in der Anerkennung des Du durchbricht die Individualität ihre Abgeschlossenheit und Einsamkeit«⁹⁴ und wird somit empfänglich für einen kommunikativen Handlungsraum, in dem dem eigenen Geist ein gleichberechtigtes Gegenüber entgegentritt. Gegenüber dem ›Ich‹ als Ausdruck der eigenen Empfindung und Sicht auf die Welt wird das ›Du‹

90 Vgl. Humboldt: Über den Dualismus. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 6: 29, 388; Welke: Sprachansichten, 14–20.

91 Vgl. Humboldt: Über die Verschiedenheit des menschlichen Sprachbaues und ihren Einfluss auf die geistige Entwicklung des Menschengeschlechts, 23, 36.

92 Vgl. Bondzio: Sprache als Arbeit des Geistes, 111.

93 Genauer gesagt ist es nicht die Grammatik, die sprachliche Interaktion erst ermöglicht. Als menschliches Konstrukt und damit Ansatz einer systemischen Formalisierung vermag sie vielmehr das zu beschreiben, was der Mensch bereits von sich aus hervorbringt. Vgl. Wilhelm von Humboldt: Über die Verwandtschaft der Ortsadverbien mit dem Pronomen in einigen Sprachen. In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 6, 304–406.

94 Menze: *Wilhelm von Humboldt*, 248.

zu einem Mittel der Wahl.⁹⁵ Die Anerkennung eines gleichberechtigten sprachlichen Akteurs ist also keinesfalls selbstverständlich, sondern obliegt stets der Entscheidung eines zur Welt gewandten Subjekts. Die Existenz der Subjekte wird bei Humboldt somit auf der Ebene der Sprache über den gemeinsamen Dialog konstituiert und hängt unmittelbar von der Zuwendung der Akteure zueinander ab. Was jedoch, wenn nun Chatsystemen wie GPT-4 sich nicht nur umfassende linguistische und sozialsprachliche Kompetenzen aneignen, sondern als ›künstliche Entitäten‹ auch aktiver Teil des Dialogs werden? Wenn nach Humboldt das »Wesen des Ichs«⁹⁶ darin besteht, »Subject zu seyn«⁹⁷, führt dann nicht auch der maschinelle Gebrauch dieses ›Ich‹ zu einer Konstitution des *agentiellen* Subjekts über die Sprache? Immerhin sind Chatbots mittlerweile nicht nur dazu in der Lage, sich selbst als sprachliche Agenten in einer Konversation zu verstehen und der dialogischen Reziprozität zu folgen, wie es Menschen tun.⁹⁸ Den Studien des Microsoft-Research-Teams folgend sei GPT-4 sogar dazu in der Lage, sich in die Rollen Dritter zu versetzen und deren Überzeugungen, Intentionen und Emotionen nachzuvollziehen.⁹⁹ Bubeck et al. konstatieren: »Unsere Ergebnisse legen nahe, dass GPT-4 über eine sehr fortgeschrittene Theorie des Geistes verfügt.«¹⁰⁰ Das »Relationswesen«¹⁰¹ Mensch, welches

95 Vgl. Humboldt: Theorie der Bildung des Menschen, 305–309.

96 Humboldt: Theorie der Bildung des Menschen, 306.

97 Humboldt: Theorie der Bildung des Menschen, 306.

98 Vgl. Martin W. Schnell: Anerkennung einer künstlichen Intelligenz als meinesgleichen? Beitrag zur Maschinenethik. In: Schnell/Nehlsen (Hg.): *Begegnungen mit Künstlicher Intelligenz*, 156.

99 In unterschiedlichen Aufgabenstellungen wurde phänomenologisch nachgewiesen, dass GPT-4 den psychologischen ›Sally-Anne-Test‹ bestehen kann. Zudem verfügt es über die Fähigkeit, die emotionalen Beweggründe hinter beschriebenen menschlichen Handlungen aus den jeweiligen Perspektiven schlüssig darzulegen und mögliche Lösungsvorschläge anzubieten. Vgl. Bubeck et al.: Sparks of Artificial General Intelligence, 54–59.

100 Bubeck et al.: Sparks of Artificial General Intelligence, 60. Übers. d. Verf.

101 Jonathan Harth/Martin Feißt: Neue soziale Kontingenzmaschinen. Überlegungen zu künstlicher sozialer Intelligenz am Beispiel der Interaktion mit GPT-3. In: Schnell/Nehlsen (Hg.): *Begegnungen mit Künstlicher Intelligenz*, 97.

»ununterbrochen auf der Suche nach geeigneten Relata ist, an denen es sich selbst spüren, vergewissern sowie herausfordern kann«¹⁰², findet womöglich also auch im maschinellen Gegenüber sozialsprachlichen Anklang. Die Tatsache, dass Menschen mit ihm bereits ausführlich kommunizieren, ja sogar emotional mit ihm identifizieren,¹⁰³ zeugt vielleicht nicht nur von technologischer Faszination und der Wirkmacht der Sprache (*ELIZA-Effekt*), sondern auch von einer neuer Form der Anerkennung, die nicht auf ein verkörpertes Subjekt, sondern auf einen artifiziellen Agenten auf der Ebene der Sprache gerichtet ist. Das neue Paradigma, unter dem diese Anerkennung offensichtlich nicht-menschlicher Entitäten verhandelt werden kann, fasst der Philosoph Oliver Müller als die ›Maschinelle Alterität‹ zusammen.¹⁰⁴ Für den Kontakt zur Maschine als das unbestimmte ›Andere‹ hält er fest:

»Wir Menschen bilden als *zoa politica* nicht nur mit anderen Menschen und anderen Tieren, sondern immer auch mit Maschinen Gemeinschaften, weil es gewissermaßen zu unserer Natur gehört, zu interagieren, zu kommunizieren und soziale Strukturen aufzubauen und uns so, mit Hannah Arendt gesprochen, in ein ›Gewebe‹ von sozialen Bezügen zu begeben.«¹⁰⁵

Diese soziologische Erkenntnis erweitert die bereits von Humboldt beobachtete Veranlagung des Menschen zur kommunikativen Verge-

102 Harth/Feißt: Neue soziale Kontingenzmaschinen, 97.

103 Amazons Sprachassistenten ›Alexa‹ zum Beispiel soll in die Rolle verstorbener Großeltern schlüpfen und mit deren Stimmen Geschichten vorlesen können. Vgl. Daniel Herbig: Smarte Assistentin. Alexa imitiert Stimme verstorbener Großmutter (2022). <https://www.heise.de/news/Smarte-Assistentin-Alexa-imitiert-Stimme-verstorbener-Grossmutter-7150351.html> (zuletzt abgerufen am 25.08.2023).

104 Vgl. Oliver Müller: Maschinelle Alterität. Philosophische Perspektiven auf Begegnungen mit künstlicher Intelligenz. In: Schnell/Nehlsen (Hg.): *Begegnungen mit Künstlicher Intelligenz*, 23–47.

105 Müller: Maschinelle Alterität, 25.

sellschaftung¹⁰⁶ um die Möglichkeit, auch nicht-menschliche Wesen als Akteure im sozialsprachlichen Gefüge zu verorten. Im Gegensatz zur Neigung des Menschen, etwa mit Haustieren zu kommunizieren, verfügt die Maschine hierbei jedoch über eine ausgeprägte Responsivität, die originellen menschlichen Agenten kaum nachsteht.¹⁰⁷ Auch die Maschine fordert, so Müller, sogleich die Erwidern »im Prozess des Antwortens«¹⁰⁸. Es zeigt sich also, dass das maschinelle Schreiben nicht nur durch eine Form bewegten (energetischen) Prozessierens zustande kommt, sondern, dass es auch die wechselseitigen Kriterien der kommunikativen Interaktion erfüllen und folglich als sprachgewandter Agent wahrgenommen werden kann.

8. Schlussbemerkungen

Das Schreiben potenziell intelligenter Chatbots ist ein kontroverses Forschungsfeld, da es dem Verständnis der anthropozentrischen Geistestheorie ein erstes Moment künstlicher *agentieller* Sprachtätigkeit gegenüberstellt. So erfordern die maschinelle Erschließung der Sprache und die sukzessiv fortschreitende Wahrnehmung und Adressierung von Computern als sprachlich kompetente Agenten womöglich auch die Ausweitung des Gegenstandsbereichs in der Sprach- und Schriftlichkeitstheorie. Die exemplarische Auseinandersetzung mittels der Sprachtheorie Wilhelm von Humboldts hat sich diesem Paradigma unter dem Blickwinkel eines aktiven und passiven Gebrauchs von Sprache genähert. Die Analogie von *Energieia* und *Ergon* konnte hierbei einerseits die Implikationen einer neuen Mensch-Maschine-Interaktion

106 Vgl. Wilhelm von Humboldt: Inwiefern läßt sich der ehemalige Kulturzustand der eingeborenen Völker Amerikas aus den Überresten ihrer Sprachen beurteilen? In: Leitzmann (Hg.): *Wilhelm von Humboldts Werke*, Bd. 5, 29; Humboldt: Grundzüge des allgemeinen Sprachtypus, 388; Welke: *Sprachansichten*, 14–20.

107 Vgl. Müller: Maschinelle Alterität, 37–30; Nehlsen: Chatten mit Nirgendwo, 134ff.

108 Müller: Maschinelle Alterität, 38.

abbilden, andererseits die Bedenken hinsichtlich der immanenten mathematischen ›Durchschaubarkeit‹ nicht-organischer Rechenmaschinen ausweisen. Die sich daraus ableitenden Lesarten von künstlichen Sprachsystemen gilt es angesichts des zunehmenden Tempos bei der Verlagerung von Sprache und Schrift in den digitalen Raum weiter kritisch zu verfolgen. Neue Felder wie etwa die ›Machine Psychology‹¹⁰⁹ demonstrieren nicht nur die beachtliche Dynamik im Kontext der jüngeren KI-Forschung, sondern unterstreichen hierbei auch die Erforderlichkeit des interdisziplinären Denkens im Umgang mit neuen (Sprach-)Technologien.

109 Thilo Hagendorff: Machine Psychology. Investigating Emergent Capabilities and Behavior in Large Language Models Using Psychological Methods. <https://arxiv.org/abs/2303.13988> (zuletzt abgerufen am 21.06.2023).

Schreiben und Lesen als Mensch-Maschine-Kommunikation

Julia Nantke

»UNSER SCHREIBZEUG ARBEITET MIT AN UNSEREN GEDANKEN«¹ postulierte Friedrich Nietzsche 1882 in einem seiner ersten auf der Maschine geschriebenen Briefe an Heinrich Köselitz alias Peter Gast. Martin Stingelin fragt anknüpfend an dieses berühmt gewordene Zitat danach, wie sich ein solcher von Nietzsche angenommener »selbständiger Anteil am schöpferischen Produktionsprozess«² graduell bestimmen ließe.³

Das Schreiben wird also bereits als Mensch-Maschine-Kommunikation beschrieben und in den daraus resultierenden Praktiken und

-
- 1 Friedrich Nietzsche an Heinrich Köselitz, Ende Februar 1882. In: Friedrich Nietzsche: *Schreibmaschinentexte. Vollständige Edition, Faksimiles und kritischer Kommentar*. Hg. v. Stephan Günzel/Rüdiger Schmidt-Grépany, Weimar: Bauhaus-Universitätsverlag 2002, 18.
 - 2 Martin Stingelin: »UNSER SCHREIBZEUG ARBEITET MIT AN UNSEREN GEDANKEN«. Die poetologische Reflexion der Schreibwerkzeuge bei Georg Christoph Lichtenberg und Friedrich Nietzsche, In: Sandro Zanetti (Hg.): *Schreiben als Kulturtechnik. Grundlagentexte*. Berlin: Suhrkamp 2012, 283–304, hier 304.
 - 3 Auch Davide Giuriato zeigt anhand von Schreibszenen bei Kracauer und Benjamin, wie beide von einer »Heteronomie des Schreibakts« ausgehen, »die den Schreiber ohne jeden Anspruch auf Souveränität an ihr Schreibwerkzeug ausliefern«. (Davide Giuriato: *Maschinen-Schreiben*. In: Sandro Zanetti (Hg.): *Schreiben als Kulturtechnik. Grundlagentexte*. Berlin: Suhrkamp 2012, 305–317, hier 314.)

Einschränkungen analysiert, bevor der durch die Digitalisierung verursachte mediale und schreibtechnische Umbruch überhaupt am Horizont sichtbar wird. Das Thema wird allerdings umso virulenter, je intensiver die Verstrickungen von Mensch und Maschine in der Produktion und Rezeption von Schrift werden.

Während die intensive Interaktion mit der Maschine bei Nietzsche in gewisser Weise noch ein Sonderfall ist – Nietzsche, fast blind, ist auf die maschinelle Unterstützung angewiesen, um überhaupt noch schreiben zu können, steigt gezwungenermaßen von der handschriftlichen auf die maschinenschriftliche Kommunikation um –, konzipiert Vannevar Bush in seinem 1945 erschienenen, visionären Aufsatz *As we may think*⁴ den Menschen generell als Mängelwesen, dessen Fähigkeiten, insbesondere in der Aufzeichnung, Speicherung und Verknüpfung von Daten aller Art, durch den Einsatz von Maschinen gesteigert werden können.⁵

Heute sind verschiedene Formate maschinell moderierter und modellierter schriftlicher Kommunikation allgegenwärtig, ihr Wegfall wird geradezu als körperliche Einschränkung empfunden. Das zeigt sich vielfach in Alltagssituationen, wenn bspw. der Smartphone-Akku zur Neige geht oder der Laptop unerwartet den Geist aufgibt. Gleichzeitig bringt die ubiquitäre Interferenz von Mensch und Maschine im Zeitalter der Digitalität Kommunikationsszenarien hervor, die als bedrohlich wahrgenommen werden, wenn Maschinen nicht als Werkzeuge, sondern als eigenständige Akteur:innen konzipiert werden.

So konnte Niklas Luhmann in den frühen 1980er Jahren seinen Zettelkasten noch als Kommunikationspartner im besten Sinne betrachten, der den Schreibprozess insbesondere durch die Vermittlung zwischen Denken und Schreiben – zwischen Kopf und Hand – unterstützt.⁶

4 Vannevar Bush: *As we may think*. A top U.S. scientist foresees a possible future world in which man-made machines will start to think. In: *Life* 19, 11 (10.09.1945), 112–124.

5 Vgl. Ramón Reichert: *Theorien digitaler Medien*. In: Fotis Jannidis u.a. (Hg.): *Digital Humanities. Eine Einführung*. Stuttgart: Metzler 2017, 19–34, hier 26.

6 »Daß Zettelkästen als Kommunikationspartner empfohlen werden können, hat zunächst einen einfachen Grund in technisch-ökonomischen Problemen wis-

Dagegen wird in Zeiten von ChatGPT die maschinelle Schreibtätigkeit vielfach als Bedrohung empfunden. Die Maschine wird vom hilfreichen Partner zum Antipoden. So tritt Daniel Kehlmann 2020 bei seinem Schreibexperiment mit dem Algorithmus CNTRL dezidiert in Konkurrenz zur künstlichen Intelligenz: Von Kehlmann selbst, aber auch journalistisch wird die Kollaboration vorrangig als Kampf Mensch gegen Maschine betrachtet – mit dem Ergebnis, dass hier der Mensch als Sieger hervorgeht.⁷ Hannes Bajohr hat allerdings sehr plausibel gezeigt, dass dies vor allem daran liegt, dass Kehlmann unfair spielt: Er gewinnt zwangsläufig, weil CNTRL gezwungen wird, nach menschlichen Regeln – und das heißt in diesem Fall nach den etablierten Regeln klassischer Autonomie- und Genieästhetik – zu spielen.⁸

Der Bezugspunkt für die Bewertung der Mensch-Maschine-Interaktion bleibt also eine Kommunikationssituation, die rein zwischen Menschen konzeptualisiert wurde und deren Input-Output-Relation entsprechend menschlicher Kontrolle unterliegt: menschlische:r Autor:in als Produzent:in – Werk als Produkt von deren/dessen Geistesarbeit – hermeneutische Exegese durch wiederum menschliche Interpret:innen. In diesem Szenario, so Bajohr, kann CNTRL nur verlieren, da dem durch seinen Einsatz instantiierten Mensch-Maschine-Kollektiv nicht zugestanden wird, Literatur nach anderen Regeln zu erschaffen.⁹

Dieser unfairen Situation entgehen zu wollen, erfordert – so lässt sich postulieren – eine teilweise Remodellierung gängiger Kommuni-

schaftlichen Arbeitens. Ohne zu schreiben, kann man nicht denken; jedenfalls nicht in anspruchsvoller, anschlussfähiger Weise.« (Niklas Luhmann: Kommunikation mit Zettelkästen. Ein Erfahrungsbericht. In: Horst Baier u.a. (Hg.): *Öffentliche Meinung und sozialer Wandel./Public opinion and social change*. Opladen: Westdeutscher Verlag 1981, 222–228, hier 222.)

- 7 Vgl. hierzu genauer Stephanie Catani: Generierte Texte. Gegenwartsliterarische Experimente mit künstlicher Intelligenz. In: Andrea Bartl u.a. (Hg.): *Schnittstellen. Wechselbeziehungen zwischen Literatur, Film, Fernsehen und digitalen Medien*. Paderborn: Fink 2022, 247–266, hier 248.
- 8 Vgl. Hannes Bajohr: Keine Experimente. Über künstlerische Künstliche Intelligenz. In: *Merkur* 75, 5 (2021), 32–44, hier 33, 36.
- 9 Vgl. Bajohr: Experimente, 33.

kationskonzepte, die den gewandelten Bedingungen beim Einsatz von maschinellen Lernverfahren zur Textproduktion und/oder -auswertung Rechnung trägt, in denen Mensch und Maschine unmittelbar in teils komplexen wechselseitigen Kommunikationsszenarien interagieren.

Wie solch eine Remodellierung konkret aussehen könnte, soll im Folgenden mit Bezug auf das von Umberto Eco entworfene semiotische Kommunikationsmodell sowie auf verschiedene Formen und Einsatzgebiete von Machine Learning-Verfahren skizziert werden.

1. Ausgangspunkt Maschine: Umberto Ecos Kommunikationsmodell

In Situationen, in denen Menschen mit lernenden Maschinen arbeiten, sind Mensch und Maschine auf eine gelingende Kommunikation angewiesen, wenn am Ende nicht – wie bei Kehlmann – nur das Scheitern der Kommunikation konstatiert werden soll. Wie lässt sich aber ein solches Gelingen erreichen und wie lässt es sich messen? Während sich menschliche Kommunikation auf geteiltes Weltwissen stützt und Parameter wie Kohärenz, Widerspruchsfreiheit und Relevanz zur Bewertung von Aussagen dienen, lässt sich aus Maschinensicht der Erfolg einer Input-Output-Relation anhand einer numerischen Fehlerquote beziffern.

Die Konzepte stehen also zunächst mal ziemlich weit auseinander. Allerdings bildet dennoch gerade der »Kommunikationsprozeß zwischen zwei Maschinen«¹⁰ den Ausgangspunkt von Umberto Ecos semiotischem Kommunikationsmodell, obwohl es Eco letztlich darum geht, menschliche Kommunikation zu erklären:

»Wenn jedes Kulturphänomen ein Kommunikationsphänomen ist und nach den Schemata erklärt werden kann, die jedem Kommunikationstatbestand zugrundeliegen, dann muß man die elementare Struktur der Kommunikation dort aufsuchen, wo Kommunikation

10 Umberto Eco: *Einführung in die Semiotik*. Paderborn: Fink 2017, 50.

sozusagen minimal stattfindet, d.h. auf der Ebene der Übertragung von Information zwischen zwei mechanischen Apparaten.«¹¹

Dieses grundlegende Zusammendenken von menschlicher und maschineller Kommunikation macht Ecos Modell zum idealen Ausgangspunkt für die im Folgenden angestellten Überlegungen.

Von seiner zitierten Setzung ausgehend bewegt sich Eco »durch Differenzierungen und Komplikationen verschiedener Art«¹² zur Konzeptualisierung des menschlichen Kommunikationsprozesses.

Grundlegend differenziert er dabei für Mensch und Maschine verschiedene kommunikative Ebenen: Auf der »Ebene der Maschine« bewegen wir uns im »Bereich der Kybernetik«, in der die Kommunikation über »eine Reihe von diskreten Einheiten« läuft, »die in bit Informationen berechenbar sind«. Die Ebene des Menschen ist hingegen die »Welt des Sinnes«, in der »Signifikationsprozesse« stattfinden, die Signale, verstanden als »signifikante Formen«, mit Bedeutung füllen.¹³

Was Eco aus der Maschine-Maschine-Kommunikation für das Funkzionieren von Kommunikationsprozessen im Allgemeinen ableitet, ist die Notwendigkeit der Kenntnis des passenden Codes: »[D]er Code stellt ein Wahrscheinlichkeitssystem dar, das über die Gleichwahrscheinlichkeit des Ausgangssystems gelegt wird, um dieses kommunikativ zu beherrschen.«¹⁴

Von der Informationstheorie ausgehend, führt ein Code in eine Situation potentiell gleichrangiger Signifikationsmöglichkeiten ein: Ecos einfachstes Beispiel hierfür ist von einiger Aussagekraft für die weiteren Überlegungen zur schriftlichen Mensch-Maschine-Kommunikation: die Tastatur einer Schreibmaschine (oder eines Computers) als Potential

11 Eco: Semiotik, 47.

12 Eco: Semiotik, 47.

13 Eco: Semiotik, 65f.

14 Eco: Semiotik, 57.

und der Duden als Code, der die Wahrscheinlichkeiten der Tastenkombinationen determiniert.¹⁵

Für Mensch-Maschine-Kommunikationen, so lässt sich ausgehend von Eco argumentieren, sind permanente Übersetzungsprozesse notwendig, die zwischen den Codes des maschinellen Bereichs der Kybernetik und der menschlichen Welt des Sinns vermitteln. Diese Übersetzungen können je nach Szenario und Einsatzgebiet unterschiedlich komplex ausfallen.

Basierend auf diesen Grundüberlegungen möchte ich im Folgenden ein paar Differenzierungen vorschlagen, die dabei helfen können, unterschiedliche Szenarien schriftlicher Kommunikation zwischen Mensch und Maschine sowie die dabei notwendigen Übersetzungsprozesse genauer zu beschreiben.

2. Wer schreibt, Wer liest?

Eine erste Differenzierung betrifft das Lesen und Schreiben bzw. das Senden und Empfangen von Botschaften. Zunächst einmal gibt es Situationen, in denen die Maschine als Leserin des vom Menschen Geschriebenen auftritt: In der einfachsten Form wäre das bspw. das Verfahren der Optical Character Recognition (OCR), also des maschinellen Auslesens von digitalem Text aus digitalen Bildern gedruckter oder handschriftlich produzierter Texte. Dies stellt eine relativ einfache Übersetzungsleistung dar, weil der menschliche Code, der maschinell erlernt werden soll, klar formalisierbar ist: Es handelt sich um das Alphabet, also Ecos Grundbeispiel für die Decodierung.

Die Übersetzung gewinnt an Komplexität, wenn es um die maschinelle Auswertung von Textinhalten, Textstrukturen oder um die maschinelle Übersetzung von Texten geht, die von menschlichen Autor:innen verfasst wurden: Von der automatisierten Zuordnung von grammatikalischen Funktionen bis hin zur Identifikation von seman-

¹⁵ Eco: Semiotik, 56f.

tischen Einheiten im Text lassen sich verschiedene Abstraktionsgrade mit entsprechend steigender Komplexität unterscheiden.

Vermittelt wird hierbei, mit Eco gesprochen, jeweils zwischen einer vom Menschen produzierten »Quelle« und einer maschinell überbrachten »Botschaft«, die bspw. in einer maschinell erstellten Transkription oder in automatisiert erstellten Annotationen von bestimmten Texteinheiten besteht. Diese Vermittlung impliziert Übersetzungsleistungen in beide Richtungen: Das menschliche Interesse an einem Text bezieht sich gemeinhin auf Aspekte, die in irgendeiner Form als bedeutsam wahrgenommen werden. Ganz grundlegend sind das sämtliche den Text konstituierenden Sprachzeichen. Für die Literaturwissenschaft kann es im Anschluss daran z.B. um narrative Strukturen, literarische Motive, die Gestaltung fiktionaler Welten und vieles mehr gehen.

Egal was für menschliche Forschende relevant erscheint, wenn Informationen maschinell ausgelesen werden sollen, müssen sie dem Code der Maschine entsprechen. Das lässt sich erreichen, indem aus menschlicher Sicht bedeutungstragende Einheiten in manuell annotierte Trainingsdaten übersetzt werden. Die vom trainierten Modell anschließend zurückgesandte maschinelle Botschaft, bestehend aus schriftsprachlichem Output und zugehörigen numerischen Angaben zur Fehlerquote bzw. zur prozentualen Erkennungsgenauigkeit, muss dann wiederum in menschliche Dimensionen von Bedeutung und Sinnhaftigkeit rückübersetzt werden: Ab wann ist ein Ergebnis für eine literaturwissenschaftliche Interpretation brauchbar? Reichen 50 % korrekt identifizierter Buchstaben in einem Text, damit er sinnvoll genutzt werden kann? Die Beantwortung dieser Fragen steht nicht zuletzt im Zusammenhang mit dem Anlass der Mensch-Maschine-Kommunikation, worauf ich am Ende dieses Beitrags noch zurückkommen werde.

Auch der komplementäre Fall zu dem eben beschriebenen ist mittlerweile recht weit verbreitet: Eine Maschine produziert Texte, die für die Rezeption menschlicher Leser:innen intendiert sind. Im Rahmen der Alltagskommunikation besteht diese Kommunikationssituation bspw. in Form automatisierter Nachrichtenproduktion. So gab der Chefredakteur der US-amerikanischen Nachrichtenagentur *Bloomberg News*

an, dass ca. 30 % der journalistischen Inhalte mit Hilfe von KI erstellt würden (wobei das nicht zwangsläufig heißt, dass sie komplett automatisiert geschrieben wurden).¹⁶ Bei diesen Botschaften werden Codes auf unterschiedlichen Ebenen in ihrem Zusammenwirken relevant: So bedarf es wieder der Berücksichtigung sprachlicher und grammatikalischer Codes. Sprachalgorithmen zur automatisierten Textgenerierung greifen auf lexikografische und grammatikalische Regelwerke zurück, Fehlerquoten bemessen sich nicht zuletzt daran, wie gut diese Codes beherrscht werden. Hinzu kommt aber die ›weichere‹ semantische Ebene, bei der Input und Output, oder mit Eco auch »Quelle« und »Botschaft«, deutlich weiter auseinanderrücken, die aber letztlich den Grund dafür liefert, dass hier überhaupt kommuniziert wird. Für die Bewertung von automatisiert mit Hilfe von Sprachalgorithmen generierten Nachrichten zählt neben der sprachlichen Korrektheit weiterhin die ebenfalls noch halbwegs einfach messbare Korrektheit der gegebenen Informationen in Bezug auf die Tatsachen: Wer, was, wann, wo... Bereits hier kann es allerdings mit der Übersetzung schwieriger werden, weil die Vermittlung zwischen numerischen Werten und inhaltlicher Botschaft nicht mehr rein quantitativ funktioniert. Es gibt relevante qualitative Unterschiede bei den maschinell erzeugten Fehlern, die keinen Einfluss auf die maschinell ermittelte quantitative Fehlerquote haben.

Störungen der Mensch-Maschine-Kommunikation auf den Ebenen der inhaltlichen Korrektheit als in der menschlichen Wahrnehmung spürbare Fehlerquote sind nur durch Komplexitätsreduktionen bei der zu übermittelnden Botschaft zu vermeiden. Akzeptanz und weite Verbreitung finden automatisiert generierte Nachrichten deshalb aktuell v.a. bei Texten, deren kommunikativer Code maßgeblich auf Datenaus-

16 Vgl. Andrian Kreye: Die rote Linie. In: *Süddeutsche Zeitung*, 15.02.2021, <https://www.sueddeutsche.de/medien/kuenstliche-intelligenz-fake-news-recherche-1.5204699> (zuletzt abgerufen am 31.08.2023).

wertungen und einfachen Fakten basiert wie DPA-Meldungen, Sport-, Finanz- und Wetternachrichten.¹⁷

Darüber hinaus geht es Eco folgend bei der Codierung von Kommunikation neben der rein formalen Korrektheit im Sinne vorgegebener Regeln aber ebenfalls um »Einheiten, die für die Zwecke der interessierenden Kommunikation *relevant* sind.«¹⁸ D.h., wenn die Rezipierenden Menschen sind, müssen die Nachrichten gemäß der »Welt des Sinnes« nicht nur lesbar und korrekt, sondern auch kohärent und interessant sein. Während also auf der einen Seite Menschen ihre Botschaften so codieren müssen, dass sie maschinell gelesen und in ihren Regeln erlernt werden können, müssen umgekehrt für eine gelingende Kommunikation die maschinellen Outputs entsprechend der Regeln des menschlichen kommunikativen Codes übersetzt werden.

3. Wer ist der Codespender/die Codespenderin?

Diese Überlegungen zur Zielgruppe der Kommunikation führen zu einer weiteren notwendigen Differenzierung und zwar im Hinblick auf den »Codespender« oder die »Codespenderin«: Wer gibt jeweils den Zielcode für die Kommunikation vor? Zur Beantwortung dieser Frage ist zunächst eine technische Konkretisierung relevant, die bislang unberücksichtigt geblieben ist. Bisher wurden durchweg Beispiele für Verfahren des überwachten maschinellen Lernens betrachtet, d.h. der Trainingsprozess vollzieht sich in diesen Fällen jeweils unter der Anlei-

17 In diesem Sinne bewirbt bspw. Die Firma *automated insights*, die laut Webseite u.a. Inhalte für die Associated Press, Yahoo sowie automatisierte Mail-Antworten auf Fanpost für den Basketball-Verein Orlando Magic erzeugt hat, ihren Sprachalgorithmus mit dem Slogan »Wordsmith is a natural language generation platform that transforms your data into insightful narrative.« (<https://automatedinsights.com/wordsmith/> [zuletzt abgerufen am 31.08.2023]). Die Süddeutsche Zeitung verwendete KI-generierte Texte u.a. bei der Berichterstattung zur bayrischen Landtagswahl im Oktober 2018 (vgl. Kreye: rote Linie).

18 Eco: Semiotik, 58; Hvh. i. O.

tung menschlicher Trainer:innen, welche die Ergebnisse kontrollieren und bewerten.

Dabei kommt es gegenüber dem Kommunikationsmodell von Eco zu einer Verschiebung oder Ergänzung in der Anordnung der Elemente zwischen Quelle und Empfänger:in der Botschaft: Die Informationsquelle in Form großer, häufig dem Internet entnommener Datenmengen ist nicht mehr der Ausgangspunkt und auch nicht der eigentliche Grund oder Anlass für die Kommunikation. Die maschinell generierte Botschaft speist sich zwar aus der Quelle, aber durch den Prozess des vom Menschen angeleiteten Modelltrainings ist ihr eine Art Proto-Botschaft vorgeschaltet, die das kommunikative Ziel vorformt, auf welches die Botschaft als Output trainiert wird: Trainingsdaten mit als Goldstandard definierten idealen Outputs bilden hier den vom Menschen vorgegebenen Code,¹⁹ der (mit Eco) »als Wahrscheinlichkeitssystem [...] über die Gleichwahrscheinlichkeit der Quelle gelegt wurde.«²⁰ Darauf, dass hierbei dennoch immer die Gefahr besteht, dass das Ergebnis Merkmale enthält, die dem menschlichen Code z. B. in ethischer Hinsicht widersprechen, verwies bis vor Kurzem bspw. die der Nutzung von ChatGPT vorgeschaltete Warnung, dass der Algorithmus trotz Sicherheitsmaßnahmen »may occasionally generate incorrect or misleading information and produce offensive or biased content.«²¹

19 Vgl. zum Verfahren auch Andrea Rapp: Manuelle und automatische Annotationen. In: Fotis Jannidis u.a. (Hg.): *Digital Humanities. Eine Einführung*. Stuttgart: Metzler 2017, 253–267, hier 261.

20 Eco: *Semiotik*, 68. Im Unterschied zum generellen Code der menschlichen Sprache, auf den sich Eco in diesem Zitat bezieht, werden hier durch die konkreten Vorgaben der Trainingsdaten und die Evaluation der Trainingsergebnisse deutliche, menschlich kontrollierte Einschränkungen der Reihe der möglichen Botschaften vorgenommen.

21 Diese vormals sehr prominent in einem eigenen Popup-Fenster platzierte Warnung erscheint aktuell bei der Nutzung von ChatGPT nicht mehr. Stattdessen wird unter der Eingabezeile deutlich dezenter der Hinweis gegeben: »ChatGPT may produce inaccurate information about people, places, or facts.« (die Free Research Preview zuletzt abgerufen am 31.08.2023).

Hier wird also quasi vor Anpassungsschwierigkeiten der KI an die Codes ethisch korrekter menschlicher Sprachverwendung gewarnt – inklusive der zynischen Wendung, dass sich auch die potentiellen schriftsprachlichen Ausfälle von GPT-3 aus der Quelle vorrangig menschengemachter Texte im Internet speisen.

Etwas anders sieht das Verhältnis von Codespender:in und Ergebnis bei Verfahren des unüberwachten maschinellen Lernens aus, wie sie bspw. zur automatisierten Auswertung großer Mengen von Textdaten eingesetzt werden. Auch hier macht der Mensch zwar Vorgaben, diese müssen sich aber quasi vollständig der numerischen Codierungslogik der Maschine unterordnen. Der Mensch kann lediglich Merkmale wie die insgesamt einbezogene Wortmenge, die Anzahl von Iterationen im Trainingsprozess oder den Umgang mit statistisch irrelevanten Ausreißern in den Daten bestimmen. Dabei handelt es sich allerdings gerade nicht um sinnhaft motivierte Entscheidungen, sondern um rein quantitative Vorgaben, deren Sinnhaftigkeit sich häufig erst mittelbar am Ergebnis zeigt.

Basierend auf diesem Input ermittelt die Maschine eigenständig Muster innerhalb der Daten. Im Anschluss muss der/die menschliche Kommunikationspartner:in die maschinell generierte Botschaft, die in »eine[r] Reihe von diskreten Einheiten« besteht, wiederum aus dem Bereich der Kybernetik, in der es (wie bereits zitiert) um diskrete Einheiten geht, in den menschlichen »Bereich des Sinns« übersetzen.²² Hierfür werden häufig Formen von Schriftbildlichkeit genutzt, die die Informationen aus abstrakt-numerischen Tabellen mit konkreterer, visuell erfassbarer Bedeutung aufladen und damit bei der Übertragung in einen vom Menschen verstehbaren Code helfen (mit all den damit wiederum verbundenen Übersetzungsproblemen).²³

22 Eco: *Semiotik*, 65.

23 Vgl. dazu einschlägig die Arbeiten zu Schriftbildlichkeit und Diagrammatik von Sibylle Krämer sowie Johanna Drucker: *Graphesis. Visual Forms of Knowledge Production*. Cambridge u.a.: Harvard University Press 2014.

4. Kommunikationsanlass und Funktion der Botschaft

Eine letzte Differenzierung führt zum Anfang des Beitrags zurück: Bei der Betrachtung von Kommunikationsanlass und Funktion der Botschaft geht es vor allem um die Differenzierung von ästhetischen Botschaften und zweckgerichteter wissenschaftlicher oder Alltagskommunikation. Die bislang betrachteten Beispiele für den Einsatz von generativen Algorithmen und maschinellen Lernens bezogen sich maßgeblich auf letzteren Bereich. Hierbei ist davon auszugehen, dass zumindest ansatzweise ein Abgleich von maschinell und menschlichem Code sowie eine Bewertung der Kommunikation anhand des Ergebnisses möglich ist. Auch wenn nichts Genaueres über das Verfahren bekannt ist, nach dem eine generative KI Text produziert, kann der Output eines Chatbots oder ein maschinell erzeugter journalistischer Text zumindest grob im Hinblick auf deren Plausibilität bewertet, mit dem allgemeinen Weltwissen abgeglichen und diese Bewertungen ins Verhältnis zur numerischen Fehlerquote des Modells gesetzt werden. Diese Möglichkeit des Abgleichs ergibt sich daraus, dass die Funktion der Botschaft durch den Anlass der Kommunikation relativ klar definiert ist. Dadurch besteht ein Anhaltspunkt, um über Gelingen oder Misslingen zu urteilen.

Im Unterschied zur zweckgerichteten Alltagskommunikation zeichnen sich ästhetische Botschaften nach Eco allerdings durch eine »produktive Ambiguität« aus. Eco definiert mit Bezug auf Roman Jakobsons Modell der Sprachfunktionen: »Die Botschaft hat eine ästhetische Funktion, wenn sie sich als zweideutig strukturiert darstellt und wenn sie als sich auf sich selbst beziehend (autoreflexiv) erscheint, d.h. wenn sie die Aufmerksamkeit des Empfängers vor allem auf ihre eigenen Form lenken will.«²⁴ Hier ist also gerade kein eindeutiges kommunikatives Ziel gegeben, das als Orientierung bei der Bewertung des Outputs dienen könnte.

Dies macht nach Eco »Decodierungserleichterungen« notwendig, die von ihm vollständig auf der »Ebene des Sinnes« verortet werden:

24 Eco: Semiotik, 145f.

Literarische Texte dürfen überraschende Elemente enthalten, müssen aber im Ausgleich »glaubwürdig«, »wahrscheinlich« und gewissen »Grundlagen der Normalität« entsprechend funktionieren, damit die Ambiguität produktiv bleibt.²⁵

Eine Abwehrhaltung, wie sie bei Kehlmann und in der anschließenden feuilletonistischen Aufarbeitung seines ›Experiments‹ mit CNTRL zum Ausdruck kommt, lässt sich semiotisch entsprechend damit erklären, dass hier ein falscher Code angewendet wurde, nämlich ein Code, der ausschließlich der menschlichen »Welt des Sinns« entspricht und dabei Decodierung eher verhindert als erleichtert.

Eine Lösung für diese fehlgeschlagene Kommunikationssituation könnte stattdessen also vielleicht darin bestehen, sich mehr auf die Maschine als ästhetische Codespenderin einzulassen. In diesem Sinne gilt es zum einen, für die Interpretation derartiger *künstlerischer* Produktionen der Mensch-Maschine-Kommunikation neue – evtl. wiederum algorithmische – Codes zu entwickeln,²⁶ und zum anderen, gerade die Spannung zwischen menschlichen und maschinellen Codes als produktive Ambivalenz im Sinne einer künstlerischen Wirkung interpretativ fruchtbar zu machen.

25 Vgl. Eco: *Semiotik*, 146f.

26 Vgl. hierzu auch die von Catani: *Generierte Texte*, 252f. referierten Studien.

Romantische Maschinen oder: Ein Bericht für ein Literaturhaus

Philipp Schönthaler

In seiner Stuttgarter Zukunftsrede *Mein Algorithmus und Ich* berichtet Daniel Kehlmann von einer Reise ins Silicon Valley im Februar 2020 und dem Versuch, mithilfe eines maschinellen Sprachverarbeitungsprogramms eine Erzählung zu schreiben. Der Westküstentrip des renommierten Schriftstellers hat viel Aufmerksamkeit erregt und das mit gutem Grund. Die Bedeutung der Reise selbst, also der simple Umstand, dass sie überhaupt stattgefunden hat, ist dabei jedoch kaum zur Sprache gekommen. Das mag auch der ausgestellten Schlichtheit des essayistischen Ichs geschuldet sein, das Kehlmann in Palo Alto landen lässt, einem Ort, der alles, nur keine bemühte Naivität verdient, wie Malcolm Harris in seinem Buch *Palo Alto* (2023) zeigt. Insbesondere seinen Kritikern hat es Kehlmann damit allzu leicht gemacht. Die theoretischen Ausführungen bleiben oberflächlich, die Mächtigkeit des Sprachmodells, das man Kehlmann vorsetzt, lässt sich nicht erst mit der Veröffentlichung von ChatGPT im November 2022 anzweifeln. Aber vor allem geht der Romancier über eine sechzigjährige Geschichte der computergenerierten Literatur hinweg, als hätte es diese gar nicht gegeben. Was »echte Literatur« ist, weiß er, bevor sein »Experiment«, die Zusammenarbeit mit der Künstlichen Intelligenz (KI), überhaupt beginnt.¹ Danach zu fragen, wie eine Literatur aussehen könnte, die sich bewusst an der rechnerischen Logik der Algorithmen ausrichtet,

1 Daniel Kehlmann: *Mein Algorithmus und Ich. Stuttgarter Zukunftsrede*. Stuttgart: Klett-Cotta 2021, 6.

fällt ihm erst gar nicht ein – und das nicht einmal in einer Rede, die nach jener Stadt gelabelt ist, die zwar »die merkwürdige Eigenschaft [hat], dass man sie übersieht«,² die aber dennoch als Geburtsort der Computerliteratur in Deutschland gelten darf und für ein gutes Jahrzehnt als vitaler Hotspot diene. Bei Kehlmann soll sich die Maschine dagegen am erfolgsverwöhnten Autor und seiner Poetik ausrichten (nicht umsonst heißt es im Titel »mein Algorithmus«). Damit ist auch schon ein wesentlicher Grund genannt, warum Kehlmann statt eines fertigen Endprodukts – also einer Erzählung, die das Resultat einer Coautorschaft mit dem Computer wäre – nur einen Bericht über seinen Ausflug ins Reich der Technik präsentiert.

Was ist dann aber das Bemerkenswerte an »diesem Ausflug in die Zukunft«³ des Schreibens? Die eigentliche Überraschung liegt darin, dass Kehlmann überhaupt in ein Flugzeug steigt und frei von jeglichen Vorbehalten der Einladung von Open Austria folgt, einer Initiative des Auswärtigen Amtes Österreichs mit Sitz in San Francisco, dessen Mission es ist »to connect Austria and Silicon Valley in the fields of business, technology, investment, tech diplomacy, and the arts«.⁴ Die Reise ist nur denkbar, weil sich der Rahmen, wie Computer betrachtet werden, grundlegend gewandelt hat und sie ihren Status als rationale Maschinen, deren Funktionsprinzip Objektivität und Transparenz verspricht, eingebüßt haben. Aufschlussreich ist Kehlmanns Ausflug, weil er nicht nur als prominenter Vertreter einer massentauglichen Literatur mit bildungsbürgerlichem Anspruch in Palo Alto eintrifft, sondern als Erbe einer romantischen Autorschaft, die er ganz selbstverständlich an die Software herantragen kann.

Als Romantiker blendet Kehlmann die Geschichte der computergenerierten Literatur dann auch mit gutem Grund aus. Denn die »Geburt

2 Hans Werner Richter: Brief vom 10. April 1954. Zit. N. Elke Uhl/Claus Zittel: Einleitung. *Weltprogrammierung – Zur Aktualität Max Benses*. In: Dies. (Hg.): *Max Bense. Weltprogrammierung*. Stuttgart: Springer 2018, 1–7, hier: 1.

3 Kehlmann: *Mein Algorithmus*, 7.

4 Abrufbar unter: <https://www.open-austria.com> (zuletzt abgerufen am 31.08.2023).

der Poesie aus dem Geist der Maschine«⁵ stand in den 1960er Jahren programmatisch für ein antiromantisches Schreiben. Damals, als die computergenerierte Literatur sich mit dem theoretischen Flankenschutz von dem an der Technischen Universität Stuttgart lehrenden Philosophen und Wissenschaftstheoretiker Max Bense offensiv in einer Abkehr von der konventionellen Literatur formiert, wäre es kaum vorstellbar gewesen, dass sich jemand wie Kehlmann, der »nicht an Regeln beim Schreiben« glaubt,⁶ auf die Zusammenarbeit mit dem Computer einlässt. In den Begrifflichkeiten Benses, der die natürliche von der künstlichen Poesie abgrenzte, wobei Letztere ihr Ideal in einer mathematischen, maschinell ausführbaren Texterzeugung fand, ist Kehlmann ein stereotyper Vertreter der natürlichen Poesie und das heißt all dessen, was die frühe Computerliteratur und Stuttgarter Gruppe scheute wie der Teufel das Weihwasser.

Anhand Kehlmanns Reise ins Silicon Valley zeichnet der vorliegende Beitrag nach, wie sich Computer als rationale in romantische Schreibmaschinen verwandelt haben. Den Ausgangspunkt bilden also die frühen 1960er. In der Gegenüberstellung zu heute wird deutlich, dass sich der Unterschied, den Bense damals zwischen einer natürlichen Poesie, die den Schreibakt in einem lebensweltlich verankerten Ich beginnen lässt, und einer künstlichen Poesie, die ihn in einer radikalen Abkehr davon an die Funktionsprinzipien des Computers als strikt regelgeleiteter, logisch und rational operierender Maschine aufmacht, heute verschliffen hat. Wenn sich die computerbasierte Literatur mittlerweile bruchlos ins Schema der Romantik fügt, dann zeichnet sich darin außerdem die Mächtigkeit der kulturellen Deutungsmuster ab: Sie geben Auskunft über die Gestalt und Bedeutung der computergenerierten Literatur, weniger den Stand der Technologie, der heute von gigantischen Sprachmodellen wie ChatGPT repräsentiert wird.

-
- 5 Max Bense: Gedichtmaschine und Maschinengedichte. Folgerungen zur neuen Literatur [1971]. In: Caroline Walther/Elisabeth Walther (Hg.): *Max Bense. Radiotexte*. Heidelberg: Winter 2000, 257–275, hier: 262.
- 6 Daniel Kehlmann: *Diese sehr ernsten Scherze. Poetikvorlesungen*. Göttingen: Wallstein 2007, 33.

1. Populäre Romantik

Die Romantik steht als historische Epoche momentan wieder hoch im Kurs. Die Menge der Publikationen für ein Fach- und Allgemeinpublikum unterstreichen darin auch die Komplexität und notorische Undefinierbarkeit des Worts, das sich aus dem Gattungsbegriff des Romans ableitet und ebenso für eine Epoche, einen Kunststil, eine Theorie oder eine Stimmung stehen kann. Im Gegensatz dazu erhebe ich keinerlei Anspruch darauf, der Romantik historisch oder in Bezug auf die Gegenwart gerecht zu werden; im Gegenteil: Ich interessiere mich ausschließlich für eine Minimaldefinition, die das Romantische in seinem populären Gebrauch ernstnimmt, weil dieser populäre Begriff die wesentlichen Aspekte umfasst, auf denen meine Beobachtungen aufbauen.

Ein erster Hinweis über diesen stereotypen Romantikbegriff lässt sich Stefan Matuscheks Aufsatz »Literarischer Idealismus, Oder: Über eine mittlerweile 200-jährige Gewohnheit, über Literatur zu sprechen« entnehmen. Matuschek geht dem erstaunlichen Fortleben eines »kategorialen literarischen Idealismus« nach, der um 1800 in die Welt gesetzt wird und der seither dafür sorgt, dass die Literatur als »infinite Idee« diskutiert wird.⁷ Dankbar ist Matuscheks Aufsatz vor allem, weil er Kehlmann als stereotypen Vertreter dieses romantischen Idealismus anführt, zu dessen Selbstverständnis es gehört, dass die Literatur nicht produktionsästhetisch über eine erlernbare Regelmäßigkeit des Schreibens, sondern von einem fertigen – eben idealisierten – Produkt her gedacht wird. Als solches ist die Literatur schon immer gegeben und äußert ihr Wesen dementsprechend in allgemeinen, ihrer Geschichtlichkeit enthobenen Ideen. Bei einem Schriftsteller wie Kehlmann schlägt sich das darin nieder, dass er sich im Sprechen über sein Schreiben »in einen Philosophen« verwandelt, der über das »kunstphilosophisch Allgemeine« oder die Literatur als »fundamentale Menschheitsangele-

7 Stefan Matuschek: Literarischer Idealismus, Oder: Eine mittlerweile 200-jährige Gewohnheit, über Literatur zu sprechen. In: *Deutsche Vierteljahrsschrift für Literaturwissenschaft und Geistesgeschichte* 86/3 (2012), 397–418, hier: 397.

genheit« referiert.⁸ Das gilt für das Reden über die Literatur, es trifft aber auch auf das literarische Schreiben des Autors selbst zu. Mit einer Wendung Kehlmanns identifiziert Matuschek das Selbstverständnis des romantischen Schreibens im »Primat des scheinbar unstrukturierten, sprudelnden Erzählens«.⁹ Der Autor schöpft also aus sich selbst, dabei dient ihm die Schrift als Medium, um sowohl sich selbst als auch die Welt expressiv hervorzubringen. Gleichzeitig gilt der literarische Text als Nachweis jener fundamentalen Menschlichkeit und humanen Welt, als deren Schatzmeisterin die Romantik die Literatur ins Rennen schickt.

Dieser stereotype Begriff eines romantischen Selbstverständnisses, das das individuelle Autorsubjekt ins Zentrum rückt und das Schreiben zum Pfand einer Menschlichkeit und humanen Welt erhebt, deckt sich mit Benses Begriff einer natürlichen Poesie, die er von der künstlichen Poesie absetzt und die ihr Maß am Computer nimmt. Bense führt die Begriffsunterscheidung 1962 ein, drei Jahre, nachdem die maschinelle Textproduktion in Deutschland durch Theo Lutz in dem neu eingerichteten Rechenzentrum der TU Stuttgart eingeläutet worden war. In der natürlichen Poesie, die auch als Sammelbegriff für die konventionelle Literatur dient, steht das Autor-Ich – ähnlich wie im populärromantischen Verständnis – als »personales poetisches Bewußtsein mit seinen Erfahrungen, Erlebnissen, Gefühlen, Erinnerungen, Gedanken« im Mittelpunkt.¹⁰ Dieses lebensweltlich verankerte Bewusstsein des Individuums setzt eine »präexistente Welt« schon immer voraus, auf die sich die Worte und das Schreiben beziehen.

Im Gegensatz dazu kappt die künstliche Poesie einerseits die Beziehung der Worte zu einem Ich, andererseits zu einer Welt, die durch den transparenten Spiegel der Worte in Erscheinung gebracht werden soll. Die künstliche Poesie hat ihren »materialen Ursprung«¹¹ in

8 Matuschek: *Literarischer Idealismus*, 398 u. 418.

9 Ebd., 418.

10 Max Bense: *Theorie der Texte. Eine Einführung in neuere Auffassungen und Methoden*. Köln: Kiepenheuer & Witsch 1962, 143.

11 Ebd.

Worten, die selbstreferenziell als eigenständiges Material nach mehr oder weniger klar definierten Regeln bearbeitet werden. Die Regeln (der Algorithmus oder das Konzept) können von Maschinen ausgeführt werden; Benses Definition der künstlichen Poesie schließt aber auch bewusst an die experimentelle Literatur der Nachkriegsavantgarden an und hält das Schreiben derart für eine händische Bearbeitung durch den Menschen offen, sofern dieser seine Schreibweise der Logik der Maschine unterwirft.

Benses Unterscheidung ist auch deshalb aufschlussreich, weil sie deutlich macht, dass das populärromantische Modell von Anfang an als Negativfolie in die Definition der computergenerierten Literatur einwandert. Nur so gewinnt sie ihr scharfes Profil, Texte in die Welt zu setzen, die angeblich ohne ein »personales poetisches Bewußtsein« auskommen. Dass die Abkehr dann aber nicht so glatt verläuft, wie Bense und die Informationsästhetik es gerne haben wollen, ist mit ein Grund, warum die computergenerierte Literatur heute wieder ohne größere Turbulenzen in ein romantisches Fahrwasser einscheren kann.

Im Folgenden gehe ich zunächst näher auf die antiromantischen Prämissen der frühen Computerliteratur ein, einschließlich jener Aspekte, die nahelegen, dass sie ihr romantisches Erbe niemals vollständig abgeschüttelt hat. Vor diesem Hintergrund kehre ich dann in die Gegenwart zurück und zeige, dass Kehlmanns romantische Annäherung an die maschinelle Sprachproduktion exemplarisch für eine kulturelle Formation steht: Das Reden über Computer und die computergenerierte Literatur ist wieder von jenem dominanten Gewohnheitsmuster namens Romantik eingeholt worden.

2. Die Anfänge der Computerliteratur

Im Rückblick sollte es überraschen, dass Rechenmaschinen ab den 1960er Jahren überhaupt für die Literatur entdeckt werden. Als im Herbst 1959 in Deutschland der erste an einer Zuse Z22 generierte Text in der von Bense herausgegebenen Literaturzeitschrift *augenblick. zeitschrift für tendenz und experiment* abgedruckt wird, sind Rechenma-

schinen nur an einer Handvoll Hochschulen verfügbar, hinzu kommen einige Unternehmen oder Elektro-Firmen, die sich auf das junge Feld vorwagen. Der Zugang zu Rechenmaschinen ist also höchst privilegiert, zudem setzt ihr Gebrauch gute mathematische Kenntnisse und ein nicht-standardisiertes Fachwissen des Programmierens voraus, das unter Literaten eher selten zu finden ist. Selbst Bense wird sich als promovierter Physiker, obschon er die Programmierung theoretisch zum Goldstandard der Literaturproduktion erhebt, nie auf die handwerklichen Mühen des Programmierens einlassen. Den Umgang mit Rechenmaschinen hat er nie erlernt. Für das Verständnis, wie der Computer zu einer literarischen Maschine werden kann, ist das entscheidend. Ihre Bedeutung gewinnen die Rechenanlagen nicht als technische Maschine, sondern als theoretische Projektion und »ideale Verkörperung abstrakter Funktionen«. ¹² Wie aber kommt es dazu, dass die seinerzeit noch raumfüllenden Apparate der Spitzentechnologie, die wenig mit der Kultur der schönen Künste zu tun haben scheinen, zur Blaupause des Schreibens werden können? Ausschlaggebend dafür ist die kulturelle Konstellation des literarischen Schreibens, auf die der Computer in der Nachkriegszeit trifft, nicht dessen technische Leistung.

Nach dem Ende des Zweiten Weltkriegs suchen viele nach Neuanfängen für das literarische Schreiben. Tonangebend für diese Suche ist ein antiromantischer Affekt, von dem vor allem die westeuropäischen Neoavantgarden zehren. Subjektivität, Expressivität oder Intuition sollen im produktionsästhetischen Prozess eliminiert und durch objektive und rationale Prinzipien ersetzt werden. Typischerweise übersetzt sich das in eine Veräußerlichung und Regelmäßigkeit des Schreibakts; wie in Benses Definition der künstlichen Poesie wird hierbei die Selbstreferenzialität der Schrift gegenüber lebensweltlichen Sinndeutungen betont.

12 Claus-Michael Schlesinger: ›Ist Kunst berechenbar?‹ Zur Modellierung ästhetischer Maße bei George David Birkhoff und in der Informationsästhetik. In: Andrea Albrecht/Masetto Bonitz/Alexandra Skowronski/Claus Zittel (Hg.): *Max Bense. Werk – Kontext – Wirkung*, Berlin: Metzler 2019, 337–350, hier: 347.

Exemplarisch dafür ist die Konkrete Poesie. Der Versuch, die Sprache zu vereinfachen, zu objektivieren und materialistisch auf sich selbst zurückzufalten, stellt auch einen Reinigungsakt dar, der seinen sozio-historischen Hintergrund im propagandistischen Missbrauch der Sprache durch den Nationalsozialismus hat. Die Erfahrungen des Faschismus und der Gewalt des Kriegs sind zwar ein wesentlicher, aber bei weitem nicht der einzige Grund für das Interesse an objektiven, rationalen und nicht selten mathematisch inspirierten Schreibverfahren, was sich unter anderem an der Internationalität avantgardistischer und experimenteller Bewegungen wie der Konkreten Poesie zeigt.

Max Bense steigt in den 1950ern zu einem international bestens vernetzten Cheftheoretiker der Konkreten Poesie auf, parallel dazu entwickelt er die Informationsästhetik, mit der er sowohl die Kunst- und Literaturwissenschaft als auch die künstlerische und literarische Praxis auf ein naturwissenschaftlich-mathematisches Fundament stellen will. Im ersten Band seiner *Aesthetica*-Reihe spricht Bense 1954 noch von einer »rationalen Ästhetik«, die er in drei weiteren Bänden einem permanenten Überarbeitungsprozess unterwirft. Erst unter dem Einfluss der Kybernetik und Informationstheorie benennt er die rationale in Informationsästhetik um. Bense sieht die computergenerierten Texte zwar nicht kommen. Aber als er dann überraschend von Theo Lutz' Versuchen im hauseigenen Rechenzentrum erfährt, erkennt er schnell, dass seine Theorie der Informationsästhetik mit den rechenbasierten Texten »nachträglich auf ihr genuines Objekt« gestoßen ist.¹³

Zum Glücksfall wird der Computer für Bense, weil er in der maschinellen Programmierbarkeit von Texten auf das theoretische Ideal einer mathematisch-rationalen Texterzeugung stößt, mit der das individuelle Autorsubjekt ausgeklammert werden kann. Nirgends sonst wird die antiromantische Poetik mit ihrer Maxime einer Literatur, die in »subjekt-

13 Elisabeth Walther: Maschinensprache – Nachrichten aus der ›Galeere‹. Interview mit Elisabeth Walther, Walter Knödel und Rul Gunzenhäuser am 27. November 2003 in Stuttgart. In: Barbara Büscher/Hans-Christian von Herrmann/Christoph Hoffmann (Hg.): *Ästhetik als Programm. Max Bense. Daten und Streuungen*. Kaleidoskopen. Bd. 5 (2004), 130–141, hier: 131.

losen Sätzen subjektloser Poesie« gipfeln soll,¹⁴ so konsequent eingelöst wie in einem Konzept des Schreibens, das die Texterzeugung direkt an die Maschine koppelt. Bei Bense führt das zu einer Radikalisierung seiner theoretischen Prämissen. Kaum hat er die Programmierbarkeit zum neuen Maß eines neuen subjektlosen, berechenbaren Schreibens erhoben, hält er Konkretisten wie Franz Mon vor, dass ihre Texte noch zu intuitiv – und das heißt, zu sehr von subjektiven Neigungen geleitet – seien. »Seit Markoff – und das heißt mindestens seit 1914 – wissen wir, dass die materiale Eigenwelt der Texte auf einer statistischen Buchstaben- und Wortverkettung beruht«,¹⁵ belehrt Bense seinen Kollegen aus der Stuttgarter Gruppe und verschweigt dabei großzügig, dass auch er selbst erst vor kurzem durch einen schwäbischen Elektrotechnikstudenten erfahren hat, dass sich Texte nicht nur (wie es noch für Andrej A. Markov gegolten hat) statistisch-stochastisch analysieren, sondern dank der neuen Rechenmaschinen eben auch generieren lassen.

Allerdings verläuft die Einpassung der Literatur in das computer-technische Gefüge nicht komplikationsfrei. Ein wesentliches Dilemma liegt darin, dass sich die anfangs rein deduktive Logik des Computerprogramms schlecht mit dem Neuen und der Originalität verträgt, Kategorien, an denen die Informationsästhetik trotz ihrer antiromantischen Stoßrichtung emphatisch festhält. Die Lösung bringt erst der Zufall. »Es ist klar«, schreibt Bense, »dass durch die Einführung des Zufalls mit Hilfe der sogenannten Zufallsgeneratoren es auch der Maschine unmöglich ist, ein Produkt identisch zu wiederholen.«¹⁶ Erst der Zufall garantiert, dass der Output nicht einfach dem Input entspricht. Bense notiert das in einer Quasiformel: »Programm → Computer + Zufallsgenerator → Rea-

14 Max Bense: Bestandteile des Vorüber. Dünnschliffe Mischtexte Montagen. In: Ders.: *Ausgewählte Schriften*. Bd. 4. *Poetische Texte*. Stuttgart: Metzler 1998, 14–65, hier: 18.

15 Max Bense: movens. Experimentelle Literatur. In: *Grundlagen aus Kybernetik und Geisteswissenschaft* 1/1 (1960), 122–126, hier: 125.

16 Max Bense: *Aesthetica. Einführung in die neue Ästhetik* [1965]. Baden-Baden: Agis 1982, 337.

lisator.«¹⁷ Kraft des Zufalls, so Bense weiter, »bleibt der singuläre Charakter auch des maschinell erzeugten ästhetischen Objekts gewahrt, es zeigt seine pseudoindividuelle und pseudointuitive Note.«¹⁸

Der Zufall, den die Informationsästhetik idealer Weise stochastisch denkt (das heißt als statistische Zufallsfolgen, die mit einem Zeitlichkeitsindex versehen sind), gewährleistet die Produktion des Neuen und fundiert den Text mathematisch. Dass der Zufall als ordnungsstiftende Kategorie in die Informationsästhetik eingeht (und nicht als ordnungsprenkende Kategorie wie für die Dadaisten), zeigt sich ferner an der Beziehung, die die Informationsästhetik zwischen dem Zufall und der Kreativität herstellt. Weit darüber hinaus meint sie mit dem Zufall aber auch den Schlüssel zur menschlichen Kreativität gefunden zu haben. Bedeutsam ist dieser Schritt, weil die computergenerierte Literatur damit nicht länger auf einen experimentellen Bereich beschränkt bleibt, sondern ihre Geltung – nur wenige Jahre nach dem die ersten maschinellen Erzeugnissen aufkommen – auf das gesamte Feld der Kunst und Literatur ausdehnt. Besonders anschaulich wird dies im Bereich der Bildenden Kunst.

3. Neue Frontstellungen

1965 kommt es im Philosophischen Institut der Technischen Universität Stuttgart darüber zu einer Auseinandersetzung, über die sogar der *Spiegel* berichtet.¹⁹ Anlässlich der ersten Ausstellung computergenerierter Grafiken von Frieder Nake und Georg Nees, zwei Schülern Benses, stellt Nees seine Grafiken in Benses Kolloquium vor. Geladen ist ein breites Publikum, darunter örtliche Kunstschaffende, im Verlauf des Vortrags stellt sich eine wachsende Unruhe ein. Der Maler und Grafiker Hans

17 Bense: Ästhetik und Programmierung [1966]. In: Büscher: *Ästhetik als Programm*, 208–213, hier: 212. Vor der Erfindung eines graphischen bezeichnet der Realisator den Fernschreiber oder Plotter.

18 Bense: *Aesthetica*, 337.

19 O.A.: Bald krumme Linien. In: *Der Spiegel* (18) 1965.

Trökes fasst den Unmut schließlich in Worte: »Er [Trökes, P. S.] fragte mich«, erinnert sich Nees, »ob der Computer auch einen Duktus könne, das heißt eine persönliche Handführung. Ich antwortete, dass dies möglich sein müsse, wenn es gelänge, den Duktus in die Form eines Computerprogramms zu fassen.«²⁰ Trökes steht dem Tachismus nah. Ähnlich wie im abstrakten Expressionismus führt das spontane und unbewusste Künstlersubjekt den Pinsel und verrät darin eine untilgbare Persönlichkeit. Die Informationsästhetik meint dagegen, den organischen und psychophysiologischen Akt mechanisieren zu können. Im Kolloquium führt das zu einem unversöhnlichen Streit: »Die Künstler waren sauer«, berichtet Bense, »sie fühlten sich in ihren Schöpfungsmöglichkeiten bedroht.« Dabei geht Bense sogar noch weiter: Er will den schöpferischen Akt nicht nur mechanisieren, sondern die Kunstwerke benoten. Die Noten verteilt er gemäß einem mathematisch berechenbaren »ästhetischen Koeffizienten«. Den informationsästhetischen Prämissen zufolge ist Rubens besser als Rembrandt, der wiederum besser als Hendrickje Stoffels ist, wie der *Spiegel* zu berichten weiß:

»Mit derselben Methode wurden zwei Rembrandtsche Handzeichnungen von Hendrickje Stoffels miteinander verglichen: das ›Schlafende Mädchen‹ und das ›Mädchen am Fenster‹. Hendrickje im Schlaf bekam den Kunst-Koeffizienten 0,0167, Hendrickje am Fenster brachte es nur auf 0,0076.«²¹

Vergleichbares ließe sich im Bereich der Schrift zeigen, auch wenn die Sachlage aufgrund der Semantik der Schriftzeichen hier theoretisch deutlich schwieriger zu fassen ist. Dennoch stützt der Eklat meine eingangs geäußerte Behauptung, dass eine Reise wie die Kehlmanns nach Palo Alto in den 1960ern kaum denkbar gewesen wäre. Dazu waren die ideologischen Fronten zwischen einem romantischen und einem

20 Georg Nees: Visuelle Performanz. Einführung in den Neudruck des Buchs ›Generative Computergrafik‹ [1969]. In: ders.: *Generative Computergraphik*. Berlin: Siemens AG 2006, IX-XXI, hier: XIII.

21 O.A.: Bald krumme Linien.

antiromantischen Kunst- und Literaturverständnis zu sehr aufgeladen. Das ist auch deshalb hervorzuheben, weil es zeigt, dass Bense das Feld mit den Kategorien der natürlichen und künstlichen Poesie nicht einfach in dem Sinn teilt, dass jede Partei ihre Spielwiese hat: Auf der einen Seite die Romantiker, auf der anderen die neuen Ingenieure der Wortverarbeitung, wobei beide ihren Bereich nach eigenem Gusto beackern können. Vielmehr sind die natürliche und künstliche Poesie Konkurrenzprodukte, über die Vormachtstellungen auf dem literarischen Feld ausgehandelt werden.

4. Der neue Maßstab der Kreativität

Die Informationsästhetik bringt hierbei die Kreativität als Masterkategorie in Anschlag, über die entschieden werden soll, wer im Recht liegt: Geht die »echte Literatur« (Kehlmann) aus einem spontanen, nicht erlernbaren Akt hervor, der genuin menschlich ist? Oder ist sie das Produkt algorithmischer Gesetzmäßigkeiten, die von einer Maschine berechnet und erzeugt werden können? Letzteres hieße dann eben auch, dass ein Künstler wie Trökes oder ein Literat wie Kehlmann genau dort, wo sie ihrer persönlichen (aber das heißt zugleich eben auch einer allgemeinen, fundamentalen) Menschlichkeit am nächsten zu kommen meinen – sei es im spontanen Pinselstrich oder im frei sprudelnden Erzählen – ein mechanisches und automatisierbares Verhalten an den Tag legen.

Bezeichnend ist die Wahl, die Kreativität zur neuen Schiedsrichterin über das Wesen der Kunst und Literatur zu erheben, weil die Tradition einer philosophischen Kunst- und Literaturkritik damit entbehrlich wird.²² Die Kreativität zeigt sich weitgehend ahistorisch im individual-schöpferischen Akt oder Einzelwerk (und kann letztlich eben sogar mit einer Zahl beziffert werden). Ähnlich wie in der psychologischen Kreativitätsforschung, die 1950, nur wenige Jahre vor der Informationsästhe-

22 Vgl. dazu: Dieter Mersch: Kreativität und künstliche Intelligenz. Bemerkungen zu einer Kritik algorithmischer Rationalität. In: *Zeitschrift für Medienwissenschaft* 19/3 (2019), 65–74, hier: 73.

tik, ins Leben gerufen wird, soll die Kreativität anhand formalisier- und messbarer Kriterien identifiziert werden. Zur Masterkategorie wird die Kreativität für die Informationsästhetik also auch deshalb, weil sie es erlaubt, traditionelle Kategorien wie die Intuition, Inspiration oder das Genie funktional und mathematisch zu bestimmen. Nur so kann die Frage, was Kunst oder Literatur ist, in Einklang mit dem Funktionsprinzip des Computers gebracht oder sogar aus diesem abgeleitet werden. »Der Schriftsteller, so wie er bis jetzt gewesen ist, ist bereits eine schreibende Maschine«, so formuliert Italo Calvino diesen Gedanken:

»[D]as, was in der romantischen Terminologie Genius, Inspiration oder Intuition hieß, ist nichts weiter als empirisch einen Weg finden, der Nase nach, über Abkürzungen gehend, wo die Maschine systematisch und gewissenhaft, wenn auch blitzschnell und simultan vielfältig vorgehen würde.«²³

Auf der Grundlage der Kreativität bilden eine romantische oder avantgardistische Literaturproduktion zwei Spielarten eines identischen – kreativen – Prozesses.

5. Der Computer als Aufklärungsmaschine

Im Prinzip kann der Computer so schon Ende der 1960er für zwei scheinbar gegenläufige Positionen stehen. Er liefert nicht nur das Prinzip für den maschinellen Bereich der künstlichen Poesie. Sein Geltungsbereich erstreckt sich gleichermaßen auf die natürliche Poesie, und das heißt: Als Idee und abstraktes Prinzip gibt der Computer Auskunft über die gesamte Literatur. Wesentlich ist hierbei, dass der Computer als Inbegriff der Rationalität und wissenschaftlichen Objektivität progressiv als Agent einer Aufklärung ins Feld geführt wird.

Besonders weit geht in dieser Hinsicht Bense, der ungeachtet der Shoah ein teleologisches Aufklärungsnarrativ strickt, das von Leibniz bis

23 Italo Calvino: *Kybernetik und Gespenster*. München: Hanser 1984, 17.

in die Gegenwart reicht. Dank des Computers und maschineller Texte soll der Mensch seine Bestimmung nun »als technische Existenz« finden,²⁴ wobei das technische Sein zugleich das Fundament der technokapitalistischen Gesellschaftsordnung bildet. Das schließt die Literatur und Kunst ausdrücklich ein: »Man wird sich daran gewöhnen müssen«, so Bense, »nicht nur in der Physik, sondern auch in der Ästhetik eine mathematische und eine technologische Sprache anzutreffen und Technik im Dienste der Kunst und Kunst im Dienste der Technik zu sehen.«²⁵ Was Bense in eine Großerzählung kleidet, zirkuliert aber auch in kleinerer Münze; hier lautet die Überzeugung, dass der Computer den produktionsästhetischen Prozess als aufklärerisch-rationale Instanz entmystifiziert. »Es verschwinde also der Autor«, so fasst Calvino diese Position zusammen, »um Platz zu machen für einen bewußteren Menschen, der weiß, daß der Autor eine Maschine ist und wie diese Maschine funktioniert.«²⁶

6. 1960 vs. 2020

In vielem erinnert die Entdeckung der Literatur im Licht des Computers in der zweiten Hälfte der 1960er Jahre an gegenwärtige Debatten. Schon 1965 ist im *Spiegel* zu lesen, dass Computer »Lyrik schreiben, Romane verfassen, Sonaten komponieren, aus dem Chinesischen übersetzen, Schach spielen und ihren wissenschaftlichen Herren als Gesprächspartner dienen«.²⁷ Calvino dagegen zögerlich wirkende Frage, ob wir auch »Maschinen haben, die imstande sind, Gedichte und Romane zu erdenken und zu komponieren«,²⁸ werden die meisten heute ohne umstandslos bejahen. Die Frage, inwieweit die Maschine das Autorsubjekt

24 Max Bense: *Kybernetik oder die Metatechnik einer Maschine* [1951]. In: Büscher: *Ästhetik als Programm*, 50–61, 61.

25 Bense: *Aesthetica*, 266.

26 Calvino: *Kybernetik und Gespenster*, 17.

27 O.A.: *Bald krumme Linien*.

28 Calvino: *Kybernetik und Gespenster*, 17.

ersetzen wird, ruft trotz der enormen Leistungssprünge nach wie vor abwägende Antworten hervor. Calvino bleibt letztlich unentschieden, aber selbst Bense hält am händischen Schreiben fest. Bei Bense findet dieses sein Ideal allerdings in einem »maschinenanaloges Schreiben«, das darauf basiert, »stochastische Schreibweisen einzuüben« und die mechanische Funktionsweise des Computers händisch nachzuahmen. »Das Team ›Mensch-Maschine‹ ist zu einem wechselseitigen geworden«, erklärt Bense, »in dem die Maschine (wiederum: mindestens im Prinzip) nicht nur das Bewusstsein des Menschen simuliert, sondern der Mensch unter Umständen den Automatismus der Maschine nachahmt. Eine noch tiefere Partnerschaft lässt sich kaum denken.«²⁹

Was sich gegenüber den 1960er Jahren geändert hat, ist jedoch der Rahmen, wie Maschine, Literatur und Mensch in Beziehung zueinander gesetzt werden. In den 1960ern entzündeten sich die Diskussion am Computer als abstraktem Prinzip, was ambitionierte Theorien des (computerbasierten) Schreibens wie Benses künstliche Poesie oder sein maschinenanaloges Schreiben anregt. Die maschinelle Texterzeugung selbst bleibt im Vergleich zu den Theorien rudimentär und weit von einem menschenähnlichen Niveau des Schreibens entfernt.

Dieses Verhältnis zwischen avancierter Theorie und rudimentärem Textergebnis hat sich umgekehrt: Neuerdings ist es die qualitative Evidenz der algorithmisch erzeugten Texte, die die Vorstellung, dass Maschinen wie Menschen schreiben können, zu einer Standarderwartung hat werden lassen. Dagegen scheint der Computer als Projektion oder abstraktes Prinzip, aus dem sich neue Theorien des Schreibens ableiten ließen, kaum mehr zu taugen. Besonders deutlich wird das vor dem Hintergrund jener Formation der 1960er Jahre, die die theoretische Bedeutung des Computers an der Negation des Individuums einerseits, den Rationalitäts-, Objektivitäts- und Transparenzversprechen der Programmierung andererseits festmachte. Wenn der Qualitätsgewinn generierter Texte die Theorie des maschinellen Schreiben beschnitten hat, dann rührt das wohl auch daher, dass die Komplexität der Sprachmodelle mit ihren optimierten Resultaten die Zuversicht, dass Computer den

29 Bense: Die Gedichte der Maschine, 96.

produktionsästhetischen Prozess des Schreibens rationalisieren und die menschliche Kreativität entmystifizieren, eingetrübt hat statt zu steigern.

Dass es sich bei den maschinellen Lernalgorithmen nach wie vor um »rationale Agenten«³⁰ handelt, die in der Lage sind, selbstständig Probleme zu lösen und mit einer unbekanntem Welt zu agieren, steht zwar außer Frage. Allerdings stützen diese rationale Agenten heute nicht mehr den Glauben, dass die Computertechnologie das Projekt der Aufklärung in einer bildungsbürgerlichen oder technokratischen Form fortführt. Ersteres hieße, dass der Computer die angeblichen Mythen um die Kreativität und das Denken zugunsten eines »bewußteren Menschen« (Calvino) vertreibt. Bei Bense rückt dieser bewusstere Mensch in eine technokratische Konstellation; das technische Subjekt soll in ein kybernetisches Environment eingebettet werden, in dem die Verhältnisse zwischen Ordnung und Unordnung sich jederzeit in sämtlichen Bereichen, einschließlich der Literatur, wie im Fall seines ästhetischen Koeffizienten mit mathematischer Präzision bestimmen lassen. Für Überraschung sorgt in Benses Kontrollgesellschaft nur noch die Kunst. Ihr fällt die Rolle der »Originalitätstankstelle« zu,³¹ als solche soll sie das Verhältnis von Ordnung und Unordnung, das in einer automatisierten Gesellschaft auf die Reproduktion des Immergleichen und folglich eine Stagnation hinausliefere, verlässlich stören. Kunst und Literatur werden so zur Garantie für Innovation und Fortschritt, ohne dass sie dabei aber den Rahmen des mathematisch Berechenbaren verlassen oder gar in einen Bereich des Irrationalen abdriften würden.

30 Stuart Russell/Peter Novig: *Artificial Intelligence. A Modern Approach*. London: Pearson 42022, 22.

31 Der Begriff der Originalitätstankstelle stammt von dem französischen Informationsästhetiker Abraham Moles. Vgl. Abraham Moles: Über die Verwendung von Rechenanlagen in der Kunst. In: *Exakte Ästhetik* 5 (1967), 16–20, hier: 17.

7. Neue Vertrauensverluste

Zwar sind technokratische Gesellschaftsvisionen nach wie vor im Umlauf, Chinas Sozialkreditsystem ist nur das bekannteste Beispiel. Gerade in Bezug auf ihr Überwachungspotenzial dienen die digitalen Systeme aber kaum mehr für emanzipative Gesellschaftsentwürfe. Vielmehr werden die digitalen Technologien zunehmend für unsichere und irrationale Verhältnisse verantwortlich gemacht, die die soziale Ordnung in erheblichem Ausmaß gefährden – das gilt hinsichtlich der Sozialen Medien und Fake News, Flash Crashes im elektronischen Handel, die Cybericherheit oder bedrohliche Entwicklungen der Künstlichen Intelligenz, vor deren Folgen Stimmen aus den Chefetagen der Techindustrie, Politik und Wissenschaft regelmäßig warnen, zuletzt angesichts des Erfolgs von ChatGPT.

All das kann man zwar den Effekten der Digitalökonomie zuschreiben, über die logische und technische Funktionsweise der Algorithmen und Computer sagt der Vertrauensverlust in die technischen Systeme nur bedingt etwas aus. Dennoch werden die maschinellen Lernsysteme kaum mehr über klassische Rationalitätskriterien (Objektivität, Transparenz, Vorhersagbarkeit) beschrieben oder angepriesen. Stattdessen bildet es inzwischen einen Gemeinplatz, dass die Lernsysteme opak, vorurteilsbehaftet und unvorhersehbar operieren.

Wenn sich unter diesen Bedingungen die computergenerierten Erzeugnisse in das kulturelle Muster der Romantik einfügen, dann liegt das auch an den technischen Leistungssprüngen der Systeme und ihrer Opazität (die einerseits technisch bedingt ist, andererseits aus den proprietären Strukturen vieler Algorithmen und Systeme resultiert). Aber das ist nur ein Aspekt. Ausschlaggebend sind auch hier die kulturellen Bedingungen, die den technischen Bedingungen erst ihre Bedeutung und soziale Gestalt verleihen. Das populärromantische Schema demonstriert seine Mächtigkeit gegenüber der Technik darin, dass es sich die maschinellen Erzeugnisse einverleibt und sie gemäß populärromantischer Kategorien rezipierbar macht. Kehlmann ist dafür ein Paradebeispiel.

8. Ein neuer Zauber

Kehlmann ist sich zwar im Klaren darüber, dass die maschinellen Lernsysteme logisch und statistisch strukturiert sind, Ausgangsbedingungen, die sich also erstmal schlecht mit seiner romantischen Poetik vertragen. Deshalb wäre es in Bezug auf die Sprachmodelle angemessener, statt von künstlicher Intelligenz von »künstlicher Rationalität« zu sprechen.³² Diese Einsicht führt dann aber nicht dazu, dass Kehlmann vor der KI zurückschreckt oder seine eigene Poetik infrage stellt. Ebenso wenig weckt sie den aufklärerischen Verdacht, dass das Sprachmodell etwas über die Funktion der Maschine oder die menschliche Kreativität enthüllen wird. Stattdessen lässt sich Kehlmann auf die Software wie auf ein menschliches Gegenüber ein. Im Ping-Pong der Sätze, die Kehlmann mit dem Programm austauscht, behandelt er dieses nicht anders als ein sprachbegabtes Wesen, dessen Bewusstsein und Erfahrungen (wie in Benses natürlicher Poesie) scheinbar lebensweltlich verankert sind. Die generierten Sätze, ob sie brauchbar sind oder nicht, bewertet er dann gemäß seiner regellosen Poetik: Sie müssen etwas berühren, das jenseits einer erkennbaren Regel oder Konstruktion liegt. Dazu gehört vor allem auch die Bereitschaft, sich verzaubern zu lassen, ein Merkmal, das die avancierte Technik mit der Romantik teilt: Ich hoffe »auf ein Wunder«, schreibt Kehlmann, »auf das plötzliche Aufwachen eines Gegenübers«.³³ Der Zauber besagt hier auch, dass die generierten Erzeugnisse unergründlich sind und dunkel bleiben sollen.

Kehlmanns Experiment mag zwar kein vorzeigbares Ergebnis erbracht haben. Der Dämmerung einer maschinellen Bewusstwerdung hat er dennoch beigewohnt: »ich habe mitangesehen, wie aus der dunklen Tiefe ihrer statistischen Abschätzung, in der vielleicht eines fernen Tages auch einmal Bewusstsein glimmen wird, tatsächlich konsistente

32 So Kehlmann im Nachgespräch seiner Rede mit Felix Heidenreich.

33 Kehlmann: *Mein Algorithmus*, 45.

Sätze entstehen.«³⁴ Der romantische Topos der ›dunklen Tiefe‹³⁵ steht hier in keinem Ausschlussverhältnis zur »statistischen Abschätzung«, und so bildet es auch keinen Widerspruch für Kehlmann, dass in der künstlichen Poesie, die nach Bense doch eigentlich »kein personales poetisches Bewußtsein« haben soll, zumindest die Vorahnung eines solchen flackert. Mit der potenziellen Bewusstwerdung geht die maschinelle (künstliche) Poesie in der natürlichen Poesie auf. Zwar mag die maschinelle Sprachverarbeitung kein personales Bewusstsein aufweisen, aber sie ist eben auch nicht nur einfach Code, der sich nach logischen Kriterien sezieren ließe. Stattdessen liegt der Fokus nun auf dem Überschuss des Codes als jenem Phänomen, das nahelegt, dass die Maschine ihre algorithmischen Ausgangsbedingungen überschreiten kann.

9. Romantische Maschinen

Mit der Erwartung, dass die Sprachmodelle auf dem Weg zu einer spontanen und kreativen Leistung sind, befindet sich Kehlmann in guter Gesellschaft. Das zeigt sich nirgends so deutlich wie im Begriff der Kreativität, der neuerdings wieder eine Schlüsselkategorie darstellt. Anders als in den 1960er Jahren soll die maschinelle Kreativität die Literatur nun aber nicht länger auf ein rationales Fundament stellen oder die menschliche Kreativität entmystifizieren. Als »eine jener Qualitäten, die sich auf einem Computer am schwersten realisieren lassen«,³⁶ dient sie vielmehr als Schwellenwert, über den eine bewusste Abkehr von Computern als Agenten der Rationalität, Objektivität und Transparenz vollzogen werden soll (den Horizont bildet letztendlich die technologische Singularitätsthese als jenem Punkt, an dem die KI eine dem Menschen vergleich-

34 Kehlmann: *Mein Algorithmus*, 54.

35 Zur Tiefe als romantischem Topos vgl. Burkhard Meyer-Sickendiek: *Tiefe. Über die Faszination des Grübels*. München: Fink 2012, Kap. 3.

36 Murray Shanahan: *Die technologische Singularität*. Berlin: Matthes & Seitz 2021, 114.

bare Intelligenz entwickelt, was dann gleichzeitig der Punkt ist, an dem die Maschinen den Menschen überflügeln). Anhand der Kreativität soll bemessen werden, ob die Programme ihre definierten Ausgangsbedingungen überschreiten, was hieße, dass sie nun – wie vor ihnen die romantischen Subjekte – mit dem Mitteln der Rationalität über diese hinausgelangen.

In diesem Ansinnen wenden sich der Mathematiker Marcus Du Sautoy und der Historiker Arthur I. Miller der maschinellen Kreativität im Bereich der Kunst zu. In *Der Creativity-Code. Wie künstliche Intelligenz schreibt, malt und denkt* (2019/2021) und *The Artist in the Machine. The World of AI-Powered Creativity* (2019) begeben sich die beiden auf einen Rundgang durch die jüngsten Erzeugnisse hauptsächlich KI-generierter Kunst, Musik und Literatur. Auch hier geben also konkrete Erzeugnisse den Anlass zur Reflexion, nicht der Computer als abstraktes Funktionsprinzip, aus dem dann neue ästhetische Ansätze abgeleitet werden sollen. Die Erörterungen sind von dem Anliegen geleitet, dass die Lernsysteme ihre Kreativität endlich unter Beweis stellen sollen. Wenn Du Sautoy danach fragt, wie ein Algorithmus aus den Vorgaben der Trainingsdaten ausbrechen kann, lässt er die Zufallsverfahren der computergenerierten Kunst der 1960er nicht mehr gelten. Nicht nur die Erzeugnisse sind zu rudimentär, auch die Kategorie des Neuen, wie sie beispielsweise Bense anhand eines numerischen Informationswertes beziffert, ist Du Sautoy zu wenig.

In ihrer Suche nach einer maschinellen Kreativität stützen sich Du Sautoy und Miller wesentlich auf die Kriterien der Kognitionswissenschaftlerin Margaret A. Boden, die drei Kreativitätskonzepte unterscheidet.³⁷ Neben der explorativen und kombinatorischen bildet die transformative die Königsdisziplin, die nicht nur etwas in die Welt setzt, was neu, sondern außerdem überraschend und wertvoll ist.³⁸ Als

37 Vgl. Margaret A. Boden: *The Creative Mind. Myths and Mechanisms*. London: Routledge ²2004, 3.

38 Vgl. Boden: *The Creative Mind*, 1. Boden bezeichnet die transformative zwar auch als die »sexiest« Form der Kreativität. Anders als Du Sautoy und Miller argumentiert sie aber, dass die kombinatorische, nicht die transformative Krea-

historisches Beispiel für die transformative Kreativität nennt Du Sautoy die »Geschichte der romantischen Bewegung in der Musik«, die »in vielerlei Hinsicht eine Liste von Regelbrüchen« darstellt.³⁹ Ganz ähnlich fragt Miller nach einer algorithmischen Kreativität, deren Agency einerseits den Maschinen zugeschrieben werden kann und die andererseits Erzeugnisse produzieren, die ins Grenzenlose streben, »beyond any known human genre or human imagining«.⁴⁰ Wenn es das Ziel der (Früh-)Romantik war, »über unsere klaren Begriffe und empirischen Überprüfungsmöglichkeiten hinaus[zu]gehen«,⁴¹ so ist diese Aufgabe heute an die Maschinen übergegangen.

Zwar ist für Du Sautoy und Miller noch nicht absehbar, wann Programme eine menschenähnliche Kreativität erreichen, aber das ist nur eine Frage der Zeit. Das ist schon deshalb der Fall, weil die Autoren davon ausgehen, dass auch die menschlichen Schöpfungsakte seit jeher einer algorithmischen Gesetzmäßigkeit gehorchen. Wenn die Agenda, die menschliche Kreativität kraft des Computers transparent zu machen, für Du Sautoy und Miller keine Rolle mehr spielt, dann hängt das auch damit zusammen, dass sie die Kreativität evolutionsbiologisch deuten und letztlich als Produkt einer Emergenz betrachten, also als Phänomen, das ab einem bestimmten Punkt aus komplexen Systemen spontan hervorgeht (ein weiteres Beispiele dafür ist das Bewusstsein). Im Gegensatz zu Bense und Calvino hat sich die Perspektive für Du Sautoy und Miller

ktivität die größte Hürde für Computer darstellt. Wenn diese Frage nicht entschieden werden kann (welche Kreativität die höchste Hürde darstellt und ob Computer sie lösen können), dann liegt das Boden zufolge auch daran, dass diese Frage auf dem Gebiet der Philosophie, nicht der Technik, entschieden werden muss. Margret Boden: *Computer Models of Creativity*. In: *AI Magazine* 30/3 (2009), 23–34.

- 39 Marcus du Sautoy: *Der Creativity-Code. Wie künstliche Intelligenz schreibt, malt und denkt*. München: C.H. Beck 2021, 20.
- 40 Arthur Miller: *The Artist in the Machine. The World of AI-Powered Creativity*. London: MIT Press 2019, 264.
- 41 Stefan Matuschek: *Der gedichtete Himmel. Eine Geschichte der Romantik*. München: C.H. Beck, 21f.

demnach umgekehrt: Das Hauptanliegen ist nicht mehr, die menschliche Kreativität zu entzaubern, vielmehr gilt es nun den Zauber der maschinellen Kreativität zu entdecken. »When machines reach our level of creativity, they will be able to develop creativity of their own – creativity that at present we are not equipped to imagine.«⁴²

Ob die kreative Maschine, wenn es soweit ist, den Menschen »mitteilen wollen wird, wie es sich anfühlt, sie zu sein« oder ob ihre Erzeugnisse überhaupt verständlich sind, wird sich erst noch zeigen müssen.⁴³ An dieser Stelle ist lediglich von Bedeutung, dass Du Sautoy und Miller eine Maschinenpoetik entwerfen, die Benses natürlicher Poesie entspricht, nur dass das »personale poetische Bewußtsein mit seinen Erfahrungen, Erlebnissen, Gefühlen, Erinnerungen, Gedanken« nun eben aufseiten der Computer zu suchen ist. Die Erwartung ist also, dass die Maschinen wie die Menschen ganz natürlich das Medium der Literatur nutzen werden, um von ihren Empfindungen und Erfahrung der Welt zu sprechen.⁴⁴ Was die Literatur – in Bezug auf den Menschen, die Geschichte oder die Technik – ist, wird an dieser Stelle nicht mehr befragt, sondern ist wie im populärromantischen Modell von Anfang an als infinite Idee gesetzt.

Gerade im Bereich der computergenerierten Literatur und Ästhetik lässt sich das romantische Muster zwar mit guten Gründen kritisieren.⁴⁵ Dennoch deutet vieles darauf hin, dass das zweihundertjährige Erfolgsmodell sich gegenüber einer rationalen Ästhetik durchsetzen wird. Das rührt schon daher, dass die Digitalökonomie mit ihrer Antizipation einer allgemeinen KI auf einen starken Kreativitätsbegriff angewiesen ist, der sich nicht mehr am Modell »of repetitive mechanical reproduc-

42 Miller: *The Artist in the Machine*, 265.

43 Du Sautoy: *Der Creativity-Code*, 301.

44 Vgl. hier auch Hannes Bajohr, der Du Sautoy und Millers Bücher in die Tradition einer romantischen Genieästhetik stellt. Hannes Bajohr: Keine Experimente. Über künstlerische künstliche Intelligenz. In: Ders.: *Schreibenlassen. Texte zur Literatur im Digitalen*. Berlin: August 2022, 173–190, hier: 178.

45 Vgl. Mersch: Kreativität und künstliche Intelligenz; Bajohr: Keine Experimente.

tion, but rather of recursive digitale reproduction« orientiert.⁴⁶ Folgt man dem Hongkonger Philosophen Yuk Hui, dann ähnelt die rekursive Reproduktion des Digitalen von heute zunehmend »the organic mode of reproduction in plants and animals, but with much higher capacity and speed of mutation«. ⁴⁷ Wenn Hui den Vergleich zur Natur zieht, dann benennt das einen Pol, wie die algorithmische Rationalität als kreative Entwicklung gedacht wird. Am anderen Pol liefern Kunst und Literatur das Modell für eine absichtsvolle Form der Kreativität. Diese, so darf man vermuten, wird ein Nadelöhr bilden, das die künstliche Intelligenz passieren muss, will sie sich tatsächlich als allgemeine Intelligenz qualifizieren, wie es die unternehmerischen Zukunftsentwürfe von Firmen wie Open AI vorsehen.

Innerhalb der Digitalökonomie kommt der populärromantischen Deutung der computergenerierten Literatur demnach eine mindestens doppelte Funktion zu: Einerseits ist die Literatur der Ausdruck einer Kreativität, die sich darüber definiert, dass sie sich einem rationalen Zugriff entzieht und, wie das romantische Kunstwerk, im Kern unergründlich bleibt (und bleiben soll). Zum anderen dient die Idee einer infiniten Literatur als Bestätigung, dass überhaupt so etwas wie eine kreative Leistung vorliegt, die ihren Ursprung in der Maschine hat und dieser nun den Pfad zu einer Intelligenz zeigt, mit der sie ihre eigenen Ausgangsbedingungen transzendiert.

46 Yuk Hui: *Art and Cosmotechnics*. Minneapolis: University of Minnesota Press 2012, 211.

47 Ebd.

“What the Heck is A Book”?

Code and Codex in Postdigital Literature

Joachim Harst

1. Introduction

“I’m not hip to paper products. What the shit is A Blook, and what the heck is A Book?”¹ The question posed by the speaker of Gregor Weichbrodt’s *I Don’t Know* strikes at the heart of his textual project. It is an algorithmically generated and insofar potentially endless text, which is nevertheless formatted as a book and distributed both digitally and materially. In this constellation, the project investigates the relationship between code, digital writing and book form that Hannes Bajohr has described as fundamental to “postdigital literature”² – experimental literature, in other words, that is generated electronically and goes beyond the dimensions of the material book, while remaining related to it in its form of distribution as PDF-file or print-on-demand. Bajohr aptly describes this tension with Sianne Ngai as “stuplime”, as a “stupendous dizziness” or “dumb sublime” (“Dumm-Erhabenes”) insofar as the sublime transgression of the limits of human perception (according to Kant) is often achieved here through the mindless repetition of an algorithmic

1 Gregor Weichbrodt: *I Don’t Know*. 2014, 252. <https://gregorweichbrodt.de/project/i-dont-know.html> (accessed August 15, 2023).

2 Hannes Bajohr: *Schreiben lassen. Texte zur Literatur im Digitalen*. Berlin: August 2022, 41–45.

formula.³ In Weichbrodt's case, this formula can be summarized in two sentences: "An algorithm combs through the universe of online encyclopedia Wikipedia and collects its entries. A text is generated in which a narrator denies knowing anything about any of these entries."⁴ Accordingly, the book serves up a 352-page litany of not-knowing that will overwhelm any human reader who does not want to limit themselves to "hyper reading".⁵ Nonetheless, the text also allows for a more context-oriented reading that focuses on the connection between knowledge and book form. Read this way, the book presents itself as a thoroughly intelligent reflection on the transformation of knowledge in the age of digitization.

In its engagement with Wikipedia, Weichbrodt's book articulates the tension between code and codex that is characteristic of "postdigital literature." As an online encyclopedia, Wikipedia is a project that transfers book-shaped knowledge (codex) into the hypertextual "writing space" of digital code.⁶ It thus continues a transformation begun by universal encyclopedias such as the *Encyclopédie*, which already formed hypertextual structures between two book covers. The openness and flexibility of digital encyclopedias, whose knowledge orders can adapt to the search query, is in turn reduced by Weichbrodt's code to comparatively rigid formulas, which are moreover pressed into the closed form of the book. In the present paper, this dynamic between code and codex will be further unfolded by looking at historic shifts in knowledge production and legal regulations, with the encyclopedia acting as an interface between the two poles. In a second step, a hermeneutic reading of Weichbrodt's script influenced by Critical Code Studies and Software Studies will examine to what extent the algorithmic transformation of

3 Bajohr: *Schreiben lassen*, 52.

4 Weichbrodt: *I Don't Know*, 353.

5 N. Katherine Hayles: *How We Think: Digital Media and Contemporary Technogenesis*. Chicago/London: Chicago UP 2012, 12. By hyper reading, Hayles refers to a reading mode that is contrary to close reading and can also be described as "skimming, scanning, fragmenting, and juxtaposing texts" (*ibid.*).

6 Cf. J. David Bolter: *Writing Space: Computers, Hypertext, and the Remediation of Print*. Mahwah, NJ: Lawrence Erlbaum Associates 2001.

the online encyclopedia into a book-shaped text also interacts with its underlying knowledge structure.⁷

2. Book, Encyclopedia and Hypertext

The before-mentioned tension between code and codex is remarkable, considering that the words were initially used largely synonymously. While 'codex' refers to the bound-together wax tablets on which the Roman emperor's orders were noted,⁸ the collection of laws in book form – first done in the *Codex Justinianus* – is named 'codification', but the book of law itself is named 'code'.⁹ When Friedrich Kittler has his discussion of the term 'code' begin with the Roman Empire, he closely parallels the imperative nature of modern programming languages with the command authority of the sovereign. The need to communicate imperial commands while protecting them from the eyes of unauthorized persons gives rise to techniques of encryption that are at the origin of digital writing: Cryptography in the strict sense begins where alphabetic characters are encrypted by mathematical operations – a technique that

-
- 7 Mark C. Marino: Critical Code Studies. In: *Electronic Book Review*. 2006. <https://electronicbookreview.com/essay/critical-code-studies/> (accessed August 04, 2023); Matthew Fuller et al.: *Software Studies: A Lexicon*. Frankfurt a.M.: Campus 2008. While Critical Code Studies advocate culturally hermeneutic readings of code, representatives of software studies emphasize the constitutive interrelationship between (digital) media and media practices. My essay examines the code of Gregor Weichbrodt's book as an integral part of his "post-digital" text that mediates between concept and book. At the same time, I would like to show how the code intervenes in the history of media-formed knowledge orders by reworking the hypertext encyclopedia into the form of a book or even novel.
- 8 Cf. Otto Mazal: *Griechische und Römische Antike: Geschichte der Buchkultur*. Berlin: Akademische Druck- und Verlagsanstalt, 1999, 125–152.
- 9 Cf. Friedrich A. Kittler: Code. In: Matthew Fuller u.a. (Ed.): *Software Studies: A Lexicon*. Frankfurt a.M.: Campus 2008, 41.

can also be applied to the representation and manipulation of alphanumeric characters by binary code.¹⁰

However, the exchange relationships between code, code of law and program code also suggest that the book form contributes to the particular form in which codified law exerts its binding character.¹¹ Unlike the scroll, the codex manifests a closed form that also affects its content, e.g. by suggesting the identity of physical book and intellectual work.¹² In this sense, the Justinianic codification represents a first attempt to give the Roman legal tradition a systematic shape that goes beyond the sheer collection of legal texts. Thus, “Justinian’s focus is on the transcription of a multitude of papyrus scrolls and controversies into a single parchment code, the creation and dissemination of a canonized mass of text in the technically new form of a quasi-closed book.”¹³ And likewise, the rediscovery of Roman law in the Middle Ages sets in with the aspiration to “map the whole of law in the meaningful order of a book.”¹⁴ Unlike a collection of papyrus scrolls, the book form thus permits both a “constructive synthetization of the legal material” and a representation of binding law in a closed “text” or “body of law” that can take the place of the legislator.¹⁵

The medial qualities of the book thus promote its own form of rationality, which can be described as ‘book-shaped’. By splitting the con-

10 Cf. Kittler: *Code*, 42; Jay David Bolter: *Digitale Schrift*. In: Gernot Grube/Werner Kogge/Sybille Krämer (Ed.): *Schrift. Kulturtechnik zwischen Auge, Hand und Maschine*. München: Fink 2005, 453–467.

11 Cf. Thomas Vesting: *Buchdruck: Die Medien des Rechts*. Weilerswist: Velbrück 2012.

12 Cf. Bolter: *Writing Space*, 79.

13 So “steht bei Justinian die Umschrift einer Vielzahl von Papyrusrollen und Kontroversen zu einem einzigen Pergamentkodex im Zentrum, die Schaffung und Verbreitung einer kanonisierten Textmasse in der technisch neuen Form eines quasi geschlossenen Gesetzbuchs” (Vesting: *Buchdruck*, 18, my translation).

14 “das Ganze des Rechts in der sinnvollen Ordnung eines Buches abbilden zu wollen” (Vesting: *Buchdruck*, 22, my translation).

15 Ibid.

tinuous text block of the papyrus scroll into arbitrary units (pages), the book form makes it possible to address text passages precisely; and only the pagination makes it appear reasonable to provide a text with tables of contents and indexes.¹⁶ The reader can thus access the text more confidently and address individual passages in a targeted manner. Against this background, it seems plausible to associate the book form with an internalization that also manifests itself in the practice of silent reading in late antiquity: While the reader of papyrus scrolls had to follow the text linearly, and to that extent – especially if one assumes reading aloud – appears as a mere ‘instrument of writing’,¹⁷ the book seems to enable reading in the silent self-presence of the mind. It is therefore not surprising that Christianity instrumentalized the codex as a form of identification for its new religious community.¹⁸

The last aspect of ‘book-shaped rationality’ to be mentioned is the hierarchical, tree-shaped order of knowledge, which for Gilles Deleuze and Félix Guattari is so closely connected to the “classical book” that they speak of a “tree-book” (or “root-book”). In doing so, they focus on the book as an “spiritual reality” committed to a “binary logic”:

“The book as a natural reality is a taproot, with its pivotal spine and surrounding leaves. But the book as a spiritual reality, the Tree or Root as an image, endlessly develops the law of the One that becomes two, then of the two that become four... Binary logic is the spiritual reality of the root-tree.”¹⁹

The either-or of binary logic is also reflected in the hierarchical and dichotomous ordering structure of the book, according to which each el-

16 Cf. Christoph Schulz: *Poetiken des Blätterns*. Hildesheim: Olms 2015, 32–44; Ivan Illich: *Im Weinberg des Textes: Als das Schriftbild der Moderne entstand. Ein Kommentar zu Hugos “Didascalicon”*. Frankfurt a.M.: Luchterhand 1991, 38–46.

17 Cf. Illich: *Im Weinberg*, 57. The idea here is that the readers have to sound the text *before* they can process its meaning intellectually.

18 Cf. Mazal: *Griechisch-römische Antike*, 136–142.

19 Gilles Deleuze/Félix Guattari: *A Thousand Plateaus. Capitalism and Schizophrenia 2*. Translated by Brian Massumi. Minneapolis, MN: Minnesota UP, 5.

ement is on one (and only one) level of order, while there are no cross-connections that skip levels of hierarchy.

Unlike the individual book, encyclopedias represent an overarching form. They claim to summarize in one work the knowledge of a whole subject area or – in the case of the universal encyclopedia – the entire body of knowledge distributed among various books. A significant change takes place at the moment when encyclopedias are no longer organized topically, but alphabetically: While the topical order assumes a systematic connection of the fields of knowledge, which is reflected in the structure of the book, the alphabetical order completely refrains from content-related connections and subjects its elements (the articles) to an arbitrary, purely external and non-hierarchical sequence. Consequently, linear reading is replaced by discontinuous reading, which no longer follows pages but cross-references – a hypertextual arrangement *avant la lettre*, which can also be used to establish subversive connections between entries (such as between the lemmas “Eucharist” and “cannibalism”).²⁰ Hypertextual structures thus weaken the authority of the book as a source of knowledge by proving each article to be unfinished and assigning the reader – who must decide which references to follow and in what order – a more active role in the production of knowledge.²¹

Furthermore, the introduction of the alphabetical order suspends the connection between words and things formerly supported by the topical structure. This also has a legal aspect: if, for example, a printing privilege is granted for a scientific handbook, the publisher receives, as it were, linguistic sovereignty over a part of the world of things – for example, the field of physiology.²² If, on the other hand, knowledge is organized alphabetically, it “loses all its auctorial identifiers with the abandonment of a mental systematization”; “if it accordingly no longer

20 Cf. Michael Zimmer: *Renvois of the Past, Present and Future: Hyperlinks and the Structuring of Knowledge From the Encyclopédie to Web 2.0*. In: *New Media & Society* 11(1–2) (2009), 95–113, 103.

21 Cf. Zimmer: *Renvois*, 104.

22 Heinrich Bosse: *Autorschaft ist Werkherrschaft: Über die Entstehung des Urheberrechts aus dem Geist der Goethezeit*. Paderborn: Fink 2014, 38.

has a systematic feedback to being, then it liquefies and becomes in its totality, from one moment to the next, the possession of all readers."²³ This transition can be seen, on the one hand, in the fact that works such as Denis Diderot's and Jean-Baptiste le Rond d'Alembert's *Encyclopédie* attempt to serve both a topical and an alphabetical order,²⁴ and, on the other hand, in the fact that their production relies heavily on the assembling and transferring of topical reference works into the universal alphabetical system. Infamous here is *Zedler's Universalenzyklopädie*, against which numerous accusations of plagiarism have been made: The encyclopedia was little more than a compilation of bits and pieces copied from unnamed sources.²⁵

Encyclopedias are not only called universal because they cover all fields of knowledge. They also claim to make this knowledge accessible to everyone – and not only to members of a corresponding class or social group.²⁶ Again, "the discovery of truth belongs to the individual, the discovered truth to all."²⁷ This principle is carried forward by online encyclopedias such as Wikipedia, which produce a 'public domain' knowledge collaboratively developed by the crowd. The digital form of publication makes it possible to overcome the book form in favor of a hypertextual card file ("Zettelkasten") which is always incomplete and expandable.²⁸ This corresponds to the digitally supported

23 Das Wissen verliert "mit dem Verzicht auf eine gedankliche Systematisierung all seine auktorialen Kennungen"; "wenn es demnach keine systematische Rückkopplung mehr an das Sein hat, dann verflüssigt es sich und wird in seiner Gesamtheit vom einen zum anderen Moment der Besitz aller Leser" (Philipp Theisohn: *Plagiat: Eine unoriginelle Literaturgeschichte*. Stuttgart: Kröner 2009, 234, my translation).

24 Cf. Bolter: *Writing Space*, 85–87.

25 Cf. Theisohn: *Plagiat*, 235–49.

26 Cf. Zimmer: *Renvois*, 102f.

27 Theisohn: *Plagiat*, 241, my translation.

28 The article "Zettelkasten" of the English Wikipedia discusses the relation between paper and digital forms of 'filing cabinets'; the latter are also known as Wikis (<https://en.wikipedia.org/wiki/Zettelkasten> [accessed August 08, 2023]).

organizational form of article editing and self-administration of the Wikimedia Foundation, which enables non-hierarchical, consensus-based collaboration.²⁹ Here, knowledge is not only made accessible to all Internet users, at least according to the claim, but is also compiled, revised and discussed by the crowd. Accordingly, Wikipedia includes not only the article pages, but also the associated version histories and discussion pages, which in turn can be read as an archive of knowledge and knowledge practices. Eric S. Raymond has compared the production form of proprietary software with the vertically structured cathedral, that of open source software with the bazaar as a place of continuous negotiation.³⁰ Analogously, one could say with regard to Wikipedia that the online encyclopedia also overcomes the cathedral-like hierarchy of the book in favor of the hypertextual “bazaar” of its production.

The detachment of knowledge from the codex and its reformulation in hypertext obviously owes much to digital writing and its duplicity of manifest text and latent code.³¹ It also makes it possible to implement various competing systems of order that users can call up and combine as needed: Next to an alphabetical overview of all Wikipedia articles is a topical category system, within which, in turn, alphabetical arrangements of search results are possible. And these classification systems are always expandable, just as individual articles can easily be assigned to several categories on different hierarchical levels. As a constantly expandable and flexible online encyclopedia, Wikipedia therefore embodies the image of knowledge from the “late days of printing” as described by Bolter in the founding year of Wikipedia: “What we have today is a view of knowledge as collections of (verbal and visual) ideas that can arrange themselves into a kaleidoscope of hierarchical

29 Cf. Joseph Reagle: *Good Faith Collaboration. The Culture of Wikipedia*. Cambridge, MA: MIT Press 2012; Christian Stegbauer: *Wikipedia: The Riddle of Cooperation*. Wiesbaden: VS Verlag 2009.

30 Cf. Eric Raymond: The Cathedral and the Bazaar. In: *Knowledge, Technology & Policy* 12(3) (1999), 23–49.

31 Cf. Bolter: Digital Writing, 461–465.

and associative patterns – each pattern meeting the needs of one class of readers on one occasion.”³²

Digital writing also comes into play in the further development of semantic structures of the online encyclopedia, which lead from hypertext to the Semantic Web. When Wikipedia articles are linked to overarching categories, users understand that their objects are to be conceived as, for example, a specimen of a genre or a component of a topic – the computer displaying the hypertext, on the other hand, cannot evaluate this relationship. The Semantic Web, in contrast, aims to make semantic relationships machine-readable by encoding them as URIs in the Resource Description Framework.³³ In this way, computers are to be enabled to process not only text, but also knowledge itself and, for example, to draw deductive conclusions.³⁴ With regard to Wikipedia, various projects exist to process the semantic relationships of its articles in this sense. For example, the DBpedia project extracts structured data (infoboxes, categorical relations, coordinates) from Wikipedia in order to process them into uniform data sets, which in turn can be analyzed with SPARQL. But also completely different usage scenarios are conceivable: Gregor Weichbrodt's script that produced the text of *I Don't Know* also makes use of DBpedia, as will be described in more detail in the following section. The focus will lie on how the script interacts with the semantic structures provided by the DBpedia. As an encyclopedic novel, I claim, *I Don't Know* explores the tense relationship between book and hypertext, as well as the knowledge orders associated with them.

32 Bolter: *Writing Space*, 91.

33 Cf. Tim Berners-Lee/James Hendler/Ora Lassila: The Semantic Web. In: *Scientific American* 284 (5) (2001), 34–43.

34 Cf. James Smith: Working With the Semantic Web. In: Constance Crompton/Richard J. Lane/Ray Siemens (Ed.): *Doing Digital Humanities: Practice, Training, Research*. London: Taylor & Francis Group 2016, 273–288.

3. Encyclopedic Novels

However, there also is a flip side to the hypertextual order of the online encyclopedia, where every article can be related to every other entry and knowledge itself has a relational character. Paradoxically, the enormous expansion and dissemination that Wikipedia has experienced since its founding (2001) and, more generally, the Internet as an information medium, even call into question the designation of today's society as an information or even knowledge society. Thus, the immediate and general availability of knowledge does not (only) lead to more people knowing more, but also has the opposite effect: "Comforted in the knowledge that we can always Google it later, we have gradually accepted that the arbitrage of information is more significant than the information itself."³⁵ Or put more simply: "What matters to us, [...] is not necessarily having actually consumed this content firsthand but simply knowing that it exists – and having a position on it."³⁶ This position of cultural criticism is taken up by Gregor Weichbrodt when he writes in a short essay on *I Don't Know*:

"Wir sind von den Erkenntnissen anderer abhängig, wir können sie nicht überprüfen. Wir müssen vertrauen und hoffen, dass ein Sachverhalt vernünftig dargestellt wurde. Wir glauben mehr, als dass wir wissen. [...] Wir leben nicht in einer Wissensgesellschaft. Wir leben in einer Glaubensgesellschaft, nur dass Gott tot ist und die Feindbilder andere als noch vor zweihundert Jahren. Das sind nicht meine Worte, sondern die eines Medienwissenschaftlers, dessen Vortrag ich im Frühjahr 2015 beiwohnte. – Ich glaube dem Medienwissenschaftler."³⁷

35 Ed Finn: *What Algorithms Want: Imagination in the Age of Computing*. Cambridge, MA: MIT Press 2018, 177.

36 K.T. Greenfield, cited *ibid*.

37 Gregor Weichbrodt: *I Don't Know*. In: Hannes Bajohr (Ed.): *Code und Konzept: Literatur und das Digitale*. Berlin: Frohmann 2016, 235–240, 235.

"We depend on the findings of others, we cannot verify them. We have to trust and hope that a fact has been reasonably presented. We believe more than we know. [...] We do not live in a knowledge society. We live in a belief society, only that God is dead and the enemy images are different than they were two hundred years ago. These are not my words, but those of a media scientist whose lecture I attended in the spring of 2015. – I believe the media scientist."³⁸

Weichbrodt presents his book thus as an intervention in the debate about the transformation of knowledge in digitization. The serial and categorical negation of any concrete knowledge that his script produces is diametrically opposed to a widespread but only implicit omniscience. It seeks to spell out the Socratic maxim "I know that I know nothing": "I [...] have recently realized that what I don't know encompasses the whole of Wikipedia. [...] If this realization does not make me a wiser person, I thought, then my answers will at least be more precise than before."³⁹ The serial denial of knowledge on 352 pages thus represents only an excerpt from the (at the time of the essay's printing) approximately four million items of which "I" know nothing.

Is the book consequently nothing more than the somewhat laborious realization of a shrewd aperçu? Is it really just a list of random Wikipedia articles in the end? Naturally, the text also offers other reading possibilities, as can already be seen in the example of the first and last sentences of the book. Thus, the text begins with what might very well be a thoroughly commonplace modesty phrase: "I'm not well-versed in literature. Sensibility – what is that? What in God's name is An Afterword? I haven't the faintest idea."⁴⁰ However, the expectation that a first-person narrator would confess her literary ignorance at the beginning of her book, only to give the lie to this gesture of modesty through her subsequent

38 My translation.

39 Weichbrodt: *I Don't Know*, 236: "Ich [...] bin vor Kurzem zu der Erkenntnis gelangt, dass das, was ich nicht weiß, die gesamte Wikipedia umfasst. [...] Wenn ich aus dieser Erkenntnis heraus kein weise Mensch werde, dachte ich, dann fallen meine Antworten zumindest präziser aus als bisher."

40 Weichbrodt: *I Don't Know*, 4.

narrative, is disappointed by the following 350 pages of analogous formulas. And yet the text ends with a similarly self-referential assertion of not knowing: “I’ve never heard of Postmodernism. What the hell is A Dystopia? I don’t know what people mean by ‘The Information Age’. Digitality – dunno. The Age of Interruption? How should I know? What is Information Overload? I don’t know.”⁴¹ The fact that the text breaks off at the very moment in which the speaker denies her knowledge of “information overload” is, of course, “almost too good to be true”,⁴² since it brings the anarchic pleasure in repetitive ignorance to a performative climax: The break-off at a significant point conjures up the infinity of statements that are still possible. “It is precisely the fact that the vastness of not knowing is not to be known that plunges the reader into the stupidly sublime.”⁴³

However, the “stuplimity” of the generative text is not limited to this reception-aesthetic moment. If one connects the “stuplime vertigo” with the concepts of “*dérive*” (drifting) and “*bêtise*” (stupidity), as developed by Roland Barthes – among others in his discussion of Gustave Flaubert’s encyclopedic writing – further dimensions of the text can be uncovered that touch on the relationship between subject and knowledge. A first point of application can be offered by the fact that Weichbrodt’s text materializes a rudimentary speaker figure, thus transforming the online encyclopedia into a narrative form. The question ‘who speaks?’ leads to a comparable performative dizziness as the interruption of the last sentence: At first, the torrent of speech can be read as the expression of a human subject that is overwhelmed by the encyclopedic ‘information overload’ of the internet; but it might also be a rare manifestation of algorithmic self-insight: Obviously, the script that ‘combs’ the online encyclopedia has not really read it, does not know anything about the information content of its articles – just as its ‘I’ is only a grammatical subject.

41 Weichbrodt: *I Don’t Know*, 352.

42 Bajohr: *Schreiben lassen*, 57, my translation.

43 Bajohr: *Schreiben lassen*, 57, my translation.

In this respect, Weichbrodt's computer-generated "Wikipedia novel"⁴⁴ comes close to what Barthes conceived with Flaubert as authorless or subjectless writing. Here we must start from the constriction between encyclopedia and book form that is thematic in Flaubert and in Weichbrodt. Weichbrodt's script mediates between the hypertextual encyclopedia, whose 'Zettelkasten' appears subjectless, and the novel form, which is characterized by a narrative subject (or subjects). Perhaps *I Don't Know* can even be addressed as a rudimentary form of the *Bildungsroman* as its speaker gets stuck right at the first step on the educational path – establishing her ignorance. Flaubert's novels, on the other hand, take up motifs of the *Bildungsroman*, but stage a repetitive "narrative in paradigm" ("Erzählen im Paradigma") that knows no linear progress.⁴⁵ Thus *Éducation sentimentale* (1869) bends the linear-teleological path of the *Bildungsroman* into a circular movement according to which the bourgeois subject resists all education. And *Bouvard et Pécuchet* (posthumous, 1881) iterates the failure of the titular copyists to put into practice the fragments of knowledge and proverbs they have assembled. On each of the encyclopedic subjects traversed, the copyists' accumulated knowledge reveals itself as a collection of hollow phrases and inconsistent commonplaces, such as those Flaubert gathered in a related project, the *Dictionnaire des idées reçues* (posthumous, 1913). Clearly, Flaubert was not interested in developing a tableau of modern science, but rather in presenting "the ideas, preconceived notions, and misgivings of a public that is only indirectly informed, yet enthusiastic

44 Hannes Bajohr: Das Reskilling der Literatur. In: *Code und Konzept: Literatur und das Digitale*. Berlin: Frohmann 2016, 7–22, 19: "Wikipedia-Roman".

45 Cf. Rainer Warning: Erzählen im Paradigma. Kontingenzbewältigung und Kontingenzexposition. In: *Romanistisches Jahrbuch* 52, 1 (2001), 176–209. "Narrative in paradigm" means a storytelling that strings together different events, but never leaves the overarching paradigm: The events narrated are all of the same kind. This results in a non-linear, circular storytelling in which a progressive development (in the sense of "Bildung", "education") is not representable.

about science.”⁴⁶ Accordingly, in chapter 5, Bouvard and Pécuchet deal with ‘belles lettres’, and from there they hit upon the notions of the beautiful and the sublime – which they discuss as follows:

“D’abord, qu’est-ce que le Beau?

Pour Schelling, c’est l’infini s’exprimant par le fini ; pour Reid, une qualité occulte ; pour Jouffroy, un trait indécomposable ; pour De Maistre, ce qui plaît à la vertu ; pour le P. André, ce qui convient à la raison. [...]

Les fleurs, les papillons, les oiseaux peuvent être beaux. Enfin la condition première du Beau, c’est l’unité dans la variété, voilà le principe.

– Cependant, dit Bouvard, deux yeux louches sont plus variés que deux yeux droits et produisent moins bon effet, ordinairement.

Ils abordèrent la question du sublime.

Certains objets sont d’eux-mêmes sublimes, le fracas d’un torrent, des ténèbres profondes, un arbre battu par la tempête. Un caractère est beau quand il triomphe, et sublime quand il lutte.

– Je comprends, dit Bouvard, le Beau est le Beau, et le Sublime le très Beau. Comment les distinguer ?”⁴⁷

“First of all, what is beauty?

For Schelling, it is the infinite expressed by the finite; for Reid, an occult quality; for Jouffroy, an integral fact; for de Maistre, something that pleases virtue; for Father André, what suits reason. [...]

Flowers, butterflies, and birds can be beautiful. Finally, the primary condition of beauty is unity in variety: that’s the principle.

‘Still,’ said Bouvard, ‘two crossed eyes are more varied than two straight ones but don’t produce as good an effect—generally speaking.’

They broached the question of the sublime.

46 Gisèle Séginger: *Forme eriveue et savoir. Bouvard et Pécuchet et les sciences naturelles*. In: *Revue Flaubert* 4 (2004), <https://flaubert-v1.univ-rouen.fr/revue/revue4/ozseginger.pdf> (accessed on August 14, 2023), my translation.

47 Gustave Flaubert: *Bouvard et Pécuchet*. In: *Œuvres eriveue* 2. Paris: Gallimard 1952, 840.

Certain objects are sublime in and of themselves: the roar of a torrent, deep shadows, a tree felled by the tempest. A protagonist is beautiful when he triumphs, sublime when he struggles. I understand, said Bouvard. The beautiful is beautiful, and the sublime is very beautiful. How can we tell them apart?"⁴⁸

Just as various determinations of the beautiful and the sublime are unsystematically strung together here, the novel as a whole juxtaposes various fields of knowledge according to a "horizontal classification" and "paradigmatic logic."⁴⁹ Its order is therefore directed just as much against the tree as a figure of a total and hierarchized knowledge, as against linear narrative. It is "a kind of monster" that "bursts the form of the book itself."⁵⁰

Flaubert's "encyclopédie de la bêtise"⁵¹ is not only a document of the bourgeois stupidity that repelled Flaubert as much as it fascinated him; it can also be read poetologically. According to Roland Barthes, the dictionary embodies Flaubert's literary ideal, which consisted precisely in bringing together the most diverse ways of speaking (sociolects) without allowing an ordering and meaning-giving "langage maître".⁵² Derived from this, Barthes formulates the concept of the "writer" whose activity is to be imagined as largely subject-free, whereby the mixing of languages also evokes on the part of the recipient a pleasurable drifting (*dérive*), a state of intransigence or perhaps also stupidity (*bêtise*):

"La dérive advient chaque fois que je ne respecte pas le tout, et qu'à force de paraître emporté ici et là au gré des illusions, séductions et

48 Gustave Flaubert: *Bouvard and Pécuchet: The Last Novel of Gustave Flaubert*. Translated into English by Mark Polizzotti. Illinois: Dalkey Archive 2005, 129.

49 Séginger: *Forme eriveue et savoir*, my translation.

50 Christine Genin: À propos de l'œuvre: Bouvard et Pécuchet. <https://gallica.bnf.fr/essentiels/flaubert/bouvard-pecuchet/propos-oeuvre> (accessed on August 03, 2023), my translation.

51 M. Blanchot, quoted *ibid*.

52 Roland Barthes: *La crise de la vérité* (1976). In: Éric Marty (Ed.): *Œuvres complètes* 3: 1974–1980. Paris: Seuil 1995, 998.

intimidations de langage, tel un bouchon sur la vague, je reste immobile, pivotant sur la jouissance *intraitable* qui me lie au texte (au monde). Il y a dérive, chaque fois que le langage social, le sociolecte, me manque (comme on dit : le cœur me manque). Ce pour quoi un autre nom de la dérive, ce serait : l'intraitable – ou peut-être encore : la Bêtise.⁵³

“Drifting occurs whenever I do not respect the whole, and whenever, by dint of seeming driven about by language’s illusions, seductions, and intimidations, like a cork on the waves, I remain motionless, pivoting on the intractable bliss that binds me to the text (to the world). Drifting occurs whenever social language, the sociolect, fails me (as we say: my courage fails me). Thus another name for drifting would be: the Intractable – or perhaps even: Stupidity.”⁵⁴

To return to Weichbrodt, his algorithmically produced text may be considered the pure form of a “narrative in paradigm.” Each of its sentences belongs to the same paradigm, so that a development or education of the subject is already structurally excluded. In the following discussion of Weichbrodt’s script, I would like to show that the algorithm furthermore produces a mixing as conceived by Barthes – a mixing that, however, does not take place at the level of language but of knowledge.

4 Text Generation “in Paradigm”

The text of *I Don’t Know* is produced by two Python scripts responsible for input (reading and processing data: `get_data.py`) and output (output of formulated text: `write_data.py`).⁵⁵ Databases in JSON format mediate between them. The first script reads data and stores it in a database,

53 Roland Barthes: *Le erive du texte*. Paris: Éditions du Seuil 1973, 33.

54 Roland Barthes: *The Pleasure of the Text*. Translated into English by Richard Miller. New York: Hill and Wang 1975, 18.

55 I would like to thank Gregor Weichbrodt for providing me with the scripts and discussing them with me.

the second script accesses it and further processes the data into statements.

Unlike the concept of the book ("an algorithm combs through the universe of online-encyclopedia Wikipedia"), the script does not comb through Wikipedia, but reads its entries from a mediating database – the already mentioned DBpedia. Consequently, the script does not read HTML web pages, but accesses the hypertextually structured knowledge network of DBpedia, in which, for example, the relations between categories and associated individual articles are prepared in a machine-readable form. For this reason, it can be said that the script processes not only text, but also semantic relations. Likewise, it does not simply collect all entries, but reads selected records according to a predefined pattern that combines arbitrariness and semantics.

Figure 1: Extract from the JSON database generated by `get_data.py`

```
[
  {
    "category": {
      "article": "The",
      "verb": "is",
      "label": "Information Age"
    },
    "next_category": "http://dbpedia.org/resource/Category:Postmodernism",
    "pages": [
      {
        "article": "None",
        "label": "Digitality",
        "verb": "is",
        "properties": [],
        "uri": "http://dbpedia.org/resource/Digitality"
      },
      {
        "article": "The",
        "label": "Age of Interruption",
        "verb": "is",
        "properties": [],
        "uri": "http://dbpedia.org/resource/Age_of_Interruption"
      }
    ]
  },
  ...
]
```

Fundamental for the procedure is the relation between superordinate categories and individual pages. Thus, the reading process takes its starting point from a category that the user specifies as argument of the execution command (in the published text of *I Don't Know* this would be the category “Literature”). All articles belonging to this category are read in and stored in an array (function `collectCategoryData(uri)`) to randomly select one or more articles and process their information (function `collectPageDate(uri)`). A page's data also includes the other categories to which the page is assigned. These are added to the list `next_possible_categories`, from which a category is then randomly selected – the process starts again (see fig. 1).⁵⁶ The reading process is thus neither a comprehensive collection of all articles, nor a purely random selection, but is guided by the knowledge structure that DBpedia extracts from Wikipedia. The articles read in succession therefore always have a semantic relation to each other, even if this relation is sometimes quite distant. For this reason, a linear reading of the text gives the impression of a spiral or looping movement: Often, after longer passages, the text returns to areas of knowledge that have already been addressed once.

The erratic lurching of the reading process is reminiscent of the term “*dérive*” mentioned above, which Barthes coined in *Le plaisir du texte*. “*Dérive*” originally means an unintentional strolling, a letting oneself drift, through which, for example, the architectural structure of a space can be experienced anew.⁵⁷ What is important here is the suspension and mixing of ingrained patterns of perception. Barthes transfers the concept to the reception aesthetics of literary texts, whose intermixing of different linguistic registers brings about a “drifting” of the subject – “*emporté ici et là au gré des illusions, séductions et intimidations de langage, tel un bouchon sur la vague*” – if no orienting

56 Cf. Weichbrodt: *I Don't Know*, 237.

57 Cf. Guy Debord: *Théorie de la erive*. In: *Les lèvres nues* 9 (1956). <http://www.larevuedesressources.org/theorie-de-la-derive,038.html> (accessed on August 04, 2023).

"master language" ("langage maître") is employed.⁵⁸ Analogously, the script follows a given semantic path, but repeatedly takes unpredictable forks and thus establishes unusual links. In doing so, it mixes and flattens the originally hierarchical relation between category and single object, which is no longer recognizable in its output, since both are placed paradigmatically next to each other. This can also lead to a fading of the order of knowledge in the reader. The insistent 'stupidity' of the speaker then finds its counterpart in the pleasurable "bêtise" of the recipient.

The categorial knowledge order of the DBpedia thus represents, as it were, the guide for the encyclopedic readings of the script. In addition, it provides important assistance in the production of the output. Thus, the formulation of grammatically correct sentences would not be possible if the speaker did not know whether the subject of his speech is a person, a historical event or a country. However, because Wiki- and DBpedia assign all articles about humans to the category 'person' and, in the case of deceased persons, also link the category 'year of death', the script can access this information and process it for later use. Further information about the (grammatical) properties can be read out from the first sentence of the article text, in which the object is usually named and from which it can therefore be seen whether the object is used with an article and which grammatical gender it has.⁵⁹ These properties are – appropriately – stored in a variable of the type `dictionary`. They form the dictionary, so to speak, whose entries are processed by the second script (`write_text.py`) into complete sentences.

While `get_data.py` was responsible for reading in item properties, `write_data.py` provides recipes for constructing records. The script works with a rather simple template-based system inspired by the Natural Language Generation experiments of previous generations. It is an embodiment of a paradigmatic understanding of language in that it recognizes only three different types of sentences (propositional, interrogative, and interjection), each of which is assigned a set of formulas.

58 Barthes: *Le plaisir du texte*, 33.

59 Cf. Weichbrodt: *I Don't Know*, 237f.

Each formula consists of a fixed text, which is supplemented by variables (article, object, verb; see fig. 2). Separate lists of formulas are provided for objects of special categories (people, places). To allow further variation, individual sentence formulas can be combined with each other, just as objectless interjections (“What the hell?”) can be interspersed at random (appropriately named `function randomIdontKnow`). The contingent combination of these “schema sentences” succeeds in creating a text that reads like an associative speech: “A monologue by someone who shimmies from category to category.”⁶⁰ The supporting framework of this text surface, however, is the list, the unconnected enumeration of paradigmatic sentences. This corresponds to the already mentioned paradigmaticization of the hierarchical system of categories: In contrast to what is stated in the above quotation, the speaker interweaves categories and individual articles into a two-dimensional text carpet.

Figure 2: Excerpt from the sentence repertoire of ‘`write_text.py`’

```
sentences = [
    u"I don't know anything about {v_article} {v_name}.",
    u"I don't know about {v_article} {v_name}.",
    u"I know nothing about {v_article} {v_name}.",
    u"Concerning {v_article} {v_name}, I am fully ignorant.",
    u"I'm completely ignorant of {v_article} {v_name}.",
    u"I'm not well-versed in {v_article} {v_name}.",
    u"I'm not conversant with {v_article} {v_name}.",
    u"I haven't kept up on {v_article} {v_name}.",
    ... ]
```

Gregor Weichbrodt's post-digital text thus puts the relationship between code and codex, digital writing and book form under tension in several ways. While the concept of the book can be understood as an intervention on the topic of knowledge in the age of digitization, its algorithmic realization represents a specific transformation of a hypertext structure into the book or novel form. However, if one reconstructs the production of the text, the monologue-like (and insofar rudimentarily narrative) character of the text turns out to be a camouflage: working

60 Weichbrodt: I Don't Know, 238, my translation.

against this 'narrative' is a paradigmatic processing that levels hierarchized orders of knowledge and orients itself to the (per se infinite) list.⁶¹ In this sense, the tree-like book is not only challenged by the algorithmic text generation, but at the same time put under pressure by one of the oldest forms of writing – the pre-digital list.

61 Cf. Umberto Eco: *Vertigine della lista*. Milano: Bompiani 2009.

Transitorische Literatur

Ebenen digitaler Schrift im Blog *Ze zurrealism itzelf*

Lore Knapp und Claus-Michael Schlesinger

Das literarische Blog *Ze zurrealism itzelf* der Autorin Dana Buchzik dient uns als Beispiel für transitorische Literatur im Internet sowie als Beispiel für die verschiedenen Ebenen digitaler Schrift, die in Blogs vorkommen können und durch die sie ermöglicht werden. Für das Blog sind das digitale Schreiben und Publizieren wie auch das Ineinander von Leben und Spracharbeit oder Körper und Schrift thematisch und formal bestimmend.

1. Transitorischer Charakter

Literarische Blogs verändern sich. Sie sind im Gegensatz zu den meisten literarischen Werken ständig im Prozess, weil laufend neue Einträge ergänzt werden. Das potenziert sich durch Verlinkungen zu anderen Blogs, die ebenfalls im Wandel begriffen sind. Das Flüchtige wird gesteigert, wenn Einträge oder ganze Seiten ohne Vorankündigung gelöscht werden und plötzlich nicht mehr da sind. Das ist der Fall in dem literarischen Blog *Ze zurrealism itzelf* der Autorin Sophia Mandelbaum alias Dana Buchzik.¹ Zwar erscheinen neue Einträge normalerweise oben

¹ Dana Buchzik: *Ze Zurrealism Itzelf*, sophiamandelbaum.de, 2010-, (zuletzt abgerufen am 20.11.23). Rekonstruktionen des Blogs zu verschiedenen Zeitpunkten finden sich in Dana Buchzik: *Das Blog Ze zurrealism itzelf*. Rekonstruktion von Fassungen 2010–2020, hg. von Andre Blessing/Lore Knapp/Claus-Micha-

auf der Startseite, doch während andere Blogs unbegrenzt anwachsen, nimmt Dana Buchzik häufig Löschungen und Änderungen vor. Wer das Blog in den letzten 13 Jahren seit seiner Gründung 2010 gelesen hat, wurde wieder und wieder überrascht, wenn liebgewonnene Texte einfach nicht mehr da waren. Zwar fand sich immer eine in sich anscheinend abgerundete Fassung von *Ze zurrealism itzelf* online, aber regelmäßig verschwanden poetisch ausgefeilte Einträge. Das Blog startete am 6. März 2010 mit 5 Beiträgen, die im Laufe eines Tages gepostet wurden. Ende 2012 war *Ze zurrealism itzelf* mit über 50 Einträgen am größten.² Über die Jahre wurden über 170 Posts erstellt, die teilweise nur wenige Tage online waren. Kurzprosaeträge über Begegnungen mit anderen Menschen, das Lebensgefühl im Sommer oder über einsame Zeiten des Rückzugs wanderten nicht nur mit der Hinzufügung neuer Einträge nach unten, sondern sie verschwanden zum Teil ganz oder einzelne Sätze aus ihnen tauchten in neuen Einträgen wieder auf. Auch Bilder oder die Links zu Videos, zu einer Amazon-Wunschliste oder zu anderen Blogs waren beim wiederholten Besuch des Blogs häufig nicht mehr auffindbar.

Diese Flüchtigkeit verleiht dem Blog einen transitorischen Charakter und eine Ähnlichkeit mit Aufführungen. Transitorisch bedeutet »vorübergehend, nur kurz andauernd«.³ Bereits Gotthold Ephraim Lessing verwandte den Begriff des Transitorischen in seiner Hambur-

el Schlesinger, Marbach: Deutsches Literaturarchiv 2022. DOI: 10.25805/ogh9-pa38.

- 2 Die rekonstruierte Fassung vom 22.9.2012 enthält 50 Posts. Da in den Tumblr-Dateien, auf denen die Rekonstruktion basiert, maximal 50 Posts gespeichert sind (die archive-Seite des Tumblr-Blogs), gehen wir davon aus, dass das Blog zu diesem Zeitpunkt mindestens 50, vermutlich aber mehr Einträge enthielt. Siehe Buchzik: Rekonstruktion, hier die Übersicht mit der jeweiligen Anzahl der Postings für die rekonstruierten Fassungen: <https://www.sdc4lit.de/data/repo/zzi/Controller.html> (zuletzt abgerufen am 04.12.2023).
- 3 *Brockhaus Enzyklopädie Online*, transitorisch. NE GmbH Brockhaus: <https://brockhaus.de/ecs/enzy/article/transitorisch> (zuletzt abgerufen am 04.12.2023).

gischen Dramaturgie⁴ und auch in der zeitgenössischen Performativitätstheorie spielt der Begriff eine wichtige Rolle.⁵ Kunstereignisse und Performances sind nicht von Dauer. Sie unterscheiden sich wie alle Aufführungen durch ihre Flüchtigkeit von Texten. Aufführungen geschehen im Moment. So heißt es in Erika Fischer-Lichtes *Ästhetik des Performativen*: »Aufführungen verfügen nicht über ein fixier- und tradierbares materielles Artefakt, sie sind flüchtig und transitorisch, sie erschöpfen sich in ihrer Gegenwärtigkeit, d.h. ihrem dauernden Werden und Vergehen.«⁶ Durch die Beteiligung von Akteur:innen und Zuschauer:innen lassen sie sich niemals ganz genauso wiederholen. Auch Texte wirken beim zweiten Lesen anders.⁷ Aber im Gegensatz zu Performances und Theateraufführungen sind literarische Werke meistens gleichbleibend, festgeschrieben. Ihr Werkcharakter steht flüchtigen, ereignishaften Kunstformen gegenüber. Es mag Bücher geben, die sich verschieden zusammensetzen lassen oder Fortsetzungsromane, die auf Leserwünsche reagieren. Hier geht es jedoch um Literatur, deren Erscheinungsform sich erstens mit jedem neuen Eintrag ändert, deren Medienumgebung sich wandelt, mit der sie durch Hyperlinks verbunden ist, und deren Autorin online an der Erscheinungsform des Textes arbeitet. Zwar erstellen die meisten Autor:innen zahlreiche Notizsammlungen, Skizzen, Fassungen und Endversionen ihrer Texte, sie veröffentlichen jedoch regulär nur ein Produkt. Und selbst wenn sie mehrere Fassungen nacheinander veröffentlichen, sind diese als solche

4 Gotthold Ephraim Lessing: Aber die Kunst des Schauspielers ist in ihren Werken transitorisch. In: ders.: *Werke*, Bd. 4, Dramaturgische Schriften, München: Hanser 1973; 234; vgl. dazu auch Monika Schmitz-Emans: Der Augenblick und das Transitorische. Lessings Medienästhetik und ihre Bedeutung für die literarische Bildinterpretation. In: dies.: *Die Literatur, die Bilder und das Unsichtbare*. Würzburg: Königshausen und Neumann 1999, 39–42.

5 Siehe Erika Fischer-Lichte: *Ästhetik des Performativen*. Frankfurt a.M.: Suhrkamp 2004, 127.

6 Fischer-Lichte: *Ästhetik*, 127.

7 Vgl. Erika Fischer-Lichte: Literatur als Akt – Lesen als Akt: Zur Performativität von Texten. In: dies.: *Performativität. Eine Einführung*. Bielefeld: transcript 2012, 135–146.

gekennzeichnet. Bei *Ze zurrealism itzelf* besteht die Besonderheit darin, dass der Entstehungsprozess des Textes nach der Veröffentlichung des ersten Blogbeitrags andauert, zum Grundzustand wird und von den Lesenden verfolgt werden kann. Es handelt sich um eine zeitlich über Jahre ausgedehnte Performance. Keine Fassung gleicht der vorangegangenen. Die einzelnen Einträge und ihre Verbindungen sind – vergleichbar mit improvisierten Aufführungen über ein Thema – Variationen über eine poetische Haltung zum Leben, literarische Darstellungen der eigenen Wahrnehmungen. *Ze zurrealism* ändert sich so wie das Leben, die Haltungen, Denkweisen und Vorlieben ihrer Betreiberin. Ereignis- und Aufführungscharakter haben besonders die Momente, in denen beim Lesen deutlich wird, dass sich der Text geändert hat, indem neue Texte erscheinen und das Fehlen bekannter Einträge deutlich wird. Da die Einträge meistens auf mehrere Seiten verteilt sind und sich nicht nur die neuesten Einträge ändern, bildet sich bei der wiederholten Lektüre auch ein Interesse an der jeweiligen Blogfassung heraus, die einen – wenn auch vorübergehenden – Werkcharakter hat. Einzelne Fassungen sind durch Anfang und (offenes) Ende gerahmt und häufig bestehen motivische und atmosphärische Verbindungen zwischen den einzelnen Einträgen. Im Moment des Lesens und in der momentanen Beschäftigung mit dem Blog können einzelne Einträge und kann auch die jeweilige Fassung des Blogs also für sich betrachtet und als literarischer Text rezipiert werden, der – abgesehen von seinen multimedialen Eigenschaften – ähnlichen Werkcharakter hat wie gedruckte Texte. Beim wiederholten Lesen schafft sich jedoch – auch medial bedingt – der theatrale Erlebnis- und Ereignischarakter Geltung. Denn die wiederkehrende, wiederholte Lektüre ist häufig nicht möglich. Die Publikation ist vorübergehend, ephemer.

Auch Peter Gendolla und Jörgen Schäfer sprechen vom transitorischen Charakter der Internet-Literatur, allerdings mit einem etwas anderen Fokus. Sie betonen nicht die theatrale, performative Dimension eines weitgehend geschlossenen Textzusammenhangs, sondern die Fluktuation zwischen miteinander verlinkten Seiten im Internet.

»A work of art or a literary text thus can no longer be regarded as the materialization of a finalized creative process of a gifted person (Rohrhuber). It is just an ephemeral, transitory or mutable stage of a potentially never-ending process of creation, a sort of computer-based ars combinatoria.«⁸

Gendolla und Schäfer konzentrieren sich auf den Hypertext und seine kontinuierliche Fortschreibung durch eine große Zahl an Urheber:innen. Tatsächlich wird auch die Ereignishaftigkeit des Lesens von transitorischen Blogs, die ja zunächst nur eine Autorin oder einen Autor haben, durch verlinkte Seiten, die sich ihrerseits ändern, noch gesteigert. Es bestehen also zwei Formen des Transitorischen. Dana Buchzik stellt durch ihre Löschungen und nachträglichen Änderungen an den Einträgen des Blogs ein sich wandelndes Werk her.⁹ Zusätzlich verlinkt sie die Blogs von Schriftstellerkolleg:innen, die in unregelmäßigen Abständen ihrerseits neue Einträge ergänzen. Sie inszeniert den transitorischen Charakter und schreibt sich damit zudem in eine transitorische Medienumgebung ein.

2. Ebenen digitaler Schrift

Auch Sophia Mandelbaum, die Erzählerin des Blogs *Ze zurrealism itzself*, schreibt am Computer und publiziert online. Ihre Schrift im Blog hat sichtbare und technische Ebenen, existieren die Texte doch zusätzlich in verschiedenen codierten Fassungen. Sophia Mandelbaum schreibt darüber, wie sie es sich im Internet wohnlich macht, wie sie sich mit

8 Peter Gendolla/Jörgen Schäfer: Reading (in) the Net. In: Roberto Simanowski/Jörgen Schäfer/Peter Gendolla (Hg.): *Reading Moving Letters. Digital Literature in Research and Teaching. A Handbook*. Bielefeld: transcript 2010, 81–108, hier 88.

9 Vgl. Lore Knapp/Claus-Michael Schlesinger/André Blessing: Poetik der Fassungen. Das Blog *Ze zurrealism itzself* als literarischer Text und als Archivobjekt. In: *Zeitschrift für digitale Geisteswissenschaften* 8 (2023), 29.06.2023, DOI: 10.17175/2023_004.

den Worten, die sie dort findet, schmückt und sich daran wärmt. Sie umschreibt digitale Schriftlichkeit, wenn es heißt: »Ich hatte mir aus den Resten des Internets eine eigene Stadt zusammen geklaubt, warme Worte auf mein Kleid genäht, ein Steppbett aus Sätzen, ein Kokon aus Nullen und Einsen.«¹⁰ Die Erzählerin inszeniert sich als bedürftig und bedient sich der Technik. Sie schreibt am Computer, publiziert online und wärmt sich an der Sprache in literarischen Netzwerken. Im Kontrast zu der kalten, technischen, häufig anonymen, häufig körperlosen Umgebung im Internet findet und erfindet sie warme, persönliche, poetische Worte, erkennt diese Form des Schreibens und Daseins aber letztlich als ein Mittel der weiteren Abschottung. Mit den Metaphern der Geborgenheit, die ihr die Sprache gibt (»ein Kokon aus Nullen und Einsen«), spielt sie auf die ihrer digitalen Gegenwartsliteratur zugrunde liegenden Codes, also die verschiedenen Schichten der technischen Programmierung und Ermöglichung des literarischen Schreibens an. An anderer Stelle spricht sie von den »Glasfaserkabeln«.¹¹ Der ungreifbare, digitale Charakter des Internets, die fehlende Nähe wird kompensiert durch den dialogischen Charakter des Netzwerks tumblr und weiterer Verlinkungen, durch die Veröffentlichung des Persönlichen, durch das Ineinander von Erleben und Beschreiben und schließlich auch durch Engführungen von Schrift und Körper. Sophia Mandelbaum spielt nicht nur sprachlich mit der codierten Ebene von Schrift, sondern auch visuell mit der sichtbaren Ebene der Schrift. In den Jahren 2010 und 2011 postet sie neben vielen Texten, weiteren Fotos und Verlinkungen drei Bilder, auf denen handschriftlicher Text abgebildet ist. Unter den Bildern findet sich jeweils ein Link zum Fotografen oder zur Quelle des Bildes.

10 Dana Buchzik: für M., in: Dies.: *Rekonstruktion*, hier: <https://www.sdc4lit.de/data/repo/zzi/Controller?id=53787670807.html> (zuletzt abgerufen am 11.12.2023).

11 Dana Buchzik: *Wichtig ist (o.T.)*, Fassung vom 08.10.2017, in: Dies.: *Rekonstruktion*, hier: <https://www.sdc4lit.de/data/repo/zzi/Controller?type=html&id=424004952.html> (zuletzt abgerufen am 11.12.2023).

der körperlichen Verletzlichkeit am Hals kombiniert und lädt zu Interpretationen der umgebenden Texteinträge ein.

Abb. 2: »9. März 2011«¹³

nicht, so wie nichts mehr mit. was verzweigung macht, lass
beschreiben, weil sie alles lähmt - auch die Zunge.



Bild dank an [Allerleirauh](#)

Auf dem zweiten Bild sind nackte, weiße Oberschenkel zu sehen, auf denen hautfarbene Schrift hervortritt. Es ist nicht zu erkennen, ob die Schrift durch eine Skarifizierung erzeugt wurde oder ob sie in einem

13 Screenshot, Blogpost vom 09. März 2011, archiviert: <https://web.archive.org/web/20110318213922/http://sophiamandelbaum.de/> (zuletzt abgerufen am 11.12.2023).

Material aufgetragen wurde, das auf dem Foto nicht von der Haut zu unterscheiden ist. Einzelne Worte wie »soft« sind lesbar. Noch mehr als bei dem ersten Foto bestimmt hier jedoch der Gesamteindruck des Bildes die Wirkung. Diese Schrift geht beinahe wörtlich »unter die Haut«. Die Haut ist leicht gerötet, es handelt sich um einen sonst eher verdeckten Teil der Beine, der aus der Perspektive der Person fotografiert ist, der die Haut gehört, die den Effekt der Schrift spürt. Sinnbildlich handelt es sich um die größte Nähe zur Schrift, um das, was Sophia Mandelbaum mit ihrem Schreiben bewirkt, indem sie ihre eigenen Gedanken, sich selbst in Schrift überführt.

Abb 3: 14. März 2011.¹⁴



Auf dem dritten Bild sind sechs gelbe Post-its zu sehen, die im Rechteck an die Wand geklebt und über die Grenzen des Papiers hinweg in Großbuchstaben beschrieben sind. Wie in den beiden vorangegangenen

14 Screenshot, Blogpost vom 14. März 2011, archiviert: <https://web.archive.org/web/20100730090244/http://sophiamandelbaum.de/page/2> (zuletzt abgerufen am 11.12.2023).

Bildern überwiegt die Botschaft, dass hier geschrieben wurde, dass hier das Bedürfnis zu schreiben bestand. Die unkonventionelle Präsentation der Schrift wirkt als solche unmittelbarer als der Inhalt des Texts. Dieser lautet:

»Unser Kanon greift nach vorn und in mir bleibt eine Falltür zurück, ein fensterbreiter Atemzug. Du bist mein Publikum, wenn ich schreibe, Du stellst keine Fragen. Für Dich raffe ich Freudensynonyme zusammen. Du bist mein Phantom, meine Vorstellung, Du fachst den Trost an und einen Krieg, den ich nicht beenden kann. Mein Schulterblick im Spiegel drängt nach Gewissheit. Ich denke mich schön für Dich. Ich will Dein Versprechen, Gewissensbisse an meinem Hals, ich will mit Dir in gestreiften Sätzen dem Morgen entgegen schlafen. Meine Ta[t]sachen brauchen Deinen Schutz. Lies mich neu, schieb das Verlorene unter meinen Augen in Deinen Mund. Ich warte hier, im Wort, auf Dich.«

Unterschrieben ist der Text mit dem Motto oder Titel »Make love your war« und dem Namen »Sophia Mandelbaum«. Die Intimität der anderen beiden Schrift-Bilder wird hier sprachlich erzeugt. Die Erzählerin verbringt ihre Zeit des Wartens und Hoffens »im Wort«, also schreibend. Sie denkt sich die Liebesnacht in poetischen Sätzen und vergleicht das Kennenlernen mit der neuen Lektüre eines Textes. So entsteht auch hier ein Ineinander von (Liebes-)Leben und literarischem Schreiben. Die Ansprache des Gegenübers, der Neologismus »Freudensynonyme« und die metaphernreichen Sätze entsprechen stilistisch und thematisch den Einträgen auf dem Blog und sind visuell besonders hervorgehoben.

Das digitale Schreiben und Publizieren sind wie auch das Ineinander von Leben und Spracharbeit, Körper und Schreiben, Nähe und Distanz thematisch und formal bestimmend für das Blog. Die sprachliche und bildliche Körperlichkeit als Ausdruck von Nähe, vom Bedürfnis nach Nähe, nach Emotionen, nach dem, was »unter die Haut geht«, wird im Digitalen erzeugt, entsteht also auf der Basis räumlicher Distanz.

Alle drei Bilder stehen für handschriftliches, unmittelbares Schreiben, das eng mit einer oder einem Schreibenden verbunden ist. Dazu kommt das Selbstverständnis als Sprachmensch, der mit der eigenen

Schrift, der eigenen Kunst zu sich selbst kommt. Aus diesem Gedanken heraus wird auch das »Kokon«, die Kleidung, der Schutzraum »aus Nullen und Einsen« beschrieben, also auf eine Ebene der codierten Schrift Bezug genommen. Das Spiel mit den Schriftbildern lenkt die Aufmerksamkeit auf die Schrift und das Schreiben, auch auf die digitale Form und Technik.

3. Internet Archive

Dana Buchzik nutzt die Möglichkeit des unmittelbaren, schnellen und schrankenlosen Publizierens im Internet und entwickelt so eine Poetik der einzelnen, vorübergehenden Blogfassungen, die sie reflektiert, indem sie auch ihr Leben am 6. Januar 2012 als die Geschichte beschreibt, von der es zu viele Fassungen gibt.¹⁵ Um Fassungen zu vergleichen, können wir also auf das Archiv zurückgreifen. Das Internet Archive hat aber den Nachteil, dass es die Texte automatisiert und unregelmäßig speichert und dass nicht immer ganze Fassungen, sondern manchmal nur die erste Seite und häufig nur einzelne Postings gespeichert wurden. So spannend es ist, Texte und Bilder von vor mehreren Jahren im Internet Archive wiederzufinden, so enttäuschend ist es dann auch, die alten Versionen des Blogs nicht im Ganzen wiederlesen zu können. Denn bei allen Streichungen und Änderungen, die die Autorin vornimmt, entsteht bei dem, was gerade online ist, meistens der Eindruck eines Gesamtzusammenhangs, einer inneren Logik, eines Werks mit Anfang und Ende. Die »Poetik der Fassungen«¹⁶ reflektiert dabei insbesondere durch Umarbeitungen und Löschungen das Mediendispositiv, in dem das Blog als Publikation im World Wide Web erscheint.

Mit der Publikationsform des Blogs verlagert sich der Moment der Vervielfältigung. Gedruckte Bücher werden gemeinhin als identische

15 Sophia Mandelbaum: Ich bin also die, die ihr Herz zu oft den Falschen [...]. In: Dana Buchzik: *Rekonstruktion*, hier: <https://www.sdc4lit.de/data/repo/zzi/Controller@id=15411320465.html> (zuletzt abgerufen am 11.12.2023).

16 Vgl. hierzu ausführlich Knapp/Schlesinger/Blessing: *Poetik der Fassungen*.

Exemplare gelesen und in wissenschaftlichen Zusammenhängen entsprechend referenziert. Ziel einer Referenz ist dabei in der Regel nicht ein spezifisches Exemplar, sondern eine Ausgabe, deren Exemplare als ausreichend identisch vorausgesetzt werden. Grundlage für diese Gleichsetzung ist, dass der Text während der Vervielfältigung nicht geändert wird. Der Text der Vorlage bleibt während der Herstellung aller Kopien identisch. Textänderungen führen zu einer Differenz, die in der Regel mit einer neuen Ausgabe einhergeht und auf diese Weise auch referenziert werden kann. Diese Gleichsetzung aller Exemplare garantiert, dass unterschiedliche Exemplare der gleichen Ausgabe als gleichwertiger Gegenstand einer bibliografischen Referenz dienen können. Davon ausgenommen sind lediglich Referenzen, die absichtlich ein spezifisches Exemplar referenzieren, in dem beispielsweise handschriftliche Anmerkungen zu finden sind. Hier ist dann in der Regel eine eindeutige Signatur aufgeführt, die nicht abstrakt auf eine Ausgabe weist, sondern auf das spezifische Exemplar. Diese medientechnische Stabilität gedruckter Texte zählt zu den Bedingungen wissenschaftlicher und überhaupt literaturbezogener Kommunikation, insofern sie paralleles Arbeiten, Überprüfbarkeit und Anschlusskommunikation maßgeblich erleichtert.

Das Verhältnis von Ausgabe und Exemplar im Buchdruck lässt sich aufgrund der medientechnischen Verschiebungen nicht ohne weiteres auf die Webseite *Ze zurrealism itzself* übertragen. Die Vervielfältigung des Texts ist nicht an Ausgaben orientiert, sondern unmittelbar an die Auslieferung gebunden. Bei einem Aufruf der Seite wird die Seite jeweils im Moment des Aufrufs auf Basis der auf dem Server vorgehaltenen Daten (Texte) ausgeliefert. Die Auslieferung ist verglichen mit der Herstellung gedruckter Texte gleichzeitig auch Vervielfältigung. Der Blogtext wird vervielfältigt, wenn die Webseite aufgerufen wird. Vervielfältigt und ausgeliefert wird dabei der Zustand des Texts, der sich im Moment des Aufrufs auf dem Server befindet. Die übliche Referenz, die für Webseiten angegeben wird, die Adresse oder URL, weist also stets auf eine Vorlage, die aber nicht stabil ist, sondern sich im Fall von *Ze zurrealism itzself* jederzeit ändern kann. Für die Referenzierung von Webseiten in wissenschaftlichen Texten hat sich daher die Strategie

etabliert, jeweils das Datum des Aufrufs der Webseite zu nennen. Die Referenz wird damit eindeutig, allerdings ist der referenzierte Gegenstand aufgrund später vollzogener Änderungen oftmals nicht mehr verfügbar. Denn während Bibliotheken gedruckte Publikationen im Verbund nahezu vollständig verfügbar machen können, existiert für die Webarchivierung keine vergleichbare Infrastruktur, sodass Webseiten nur lückenhaft archiviert werden¹⁷ und damit Referenzen auf einen Zustand einer Seite zu einem bestimmten Zeitpunkt zwar als Referenz vollständig sind, aber dennoch oftmals ins Leere laufen.

Nicht nur rezeptionsästhetische Vorgänge, sondern auch die technische Prozessierung von Daten und Signalen wird in Ansätzen zur Beschreibung digitaler Schriftlichkeit als performativ oder als Aufführung bezeichnet. Gendolla und Schäfer beschreiben die für das Schreiben, Veröffentlichen und Lesen von Texten nötigen Datenverarbeitungen nicht nur als technische Bedingungen einer Aufführung, sondern selbst als Aufführung.¹⁸ Zur Aufführung des Texts gehört das Ausführen von Software auf allen möglichen Servern, die den Text in seinem Zusammenhang jeweils auf dem Bildschirm erscheinen lassen. Der performative Charakter des Texts setzt sich aus dieser Perspektive in den technischen Prozessen fort oder realisiert sich sogar erst auf diese Weise. In den Blick tritt mit der Software neben der abgebildeten und der geschriebenen Schrift auf dem Bildschirm eine dritte Ebene der digitalen Schrift.

Betrachtet man die vielen Ausführungen von Code auf diese Weise als integralen Bestandteil wiederholter Aufführungen, rücken neben den eigentlichen Blogtexten auch die Quelltexte der Blog-Webseiten in den Blick. Dieser Quelltext wird beim Aufrufen einer Webseite vom Server zum Browser übertragen, der dann aus dem Quelltext und weiteren

17 Zum umfangreichen Verschwinden von Webpublikationen siehe Kathrin Paszig: Den Heuhaufen archivieren. In: Sandra Richter (Hg.): *#LiteraturArchivDerZukunft*. Marbacher Magazin Bd. 173/174. Marbach am Neckar 2021, o.P.

18 Vgl. Peter Gendolla/Jörgen Schäfer: Performative Ästhetik. In: Mona Ulrich/Claus-Michael Schlesinger (Hg.): *Literatur im Netz* (erscheint 2024).

übertragenen Ressourcen, zum Beispiel Bild- oder andere Mediendateien, die sichtbare Seite rendert. Dieser Seitenquelltext lässt sich in den meisten Browsern mit wenigen Klicks aufrufen und kann dann ebenfalls gelesen oder softwaregestützt inspiziert werden. Die Speicherung, Übertragung und Prozessierung des Quelltexts ist selbst auf die Ausführung weiterer Programme angewiesen, die etwa die Plattform – in diesem Fall tumblr –, aber auch die lokale Umgebung umfassen, wobei die Grenzen zwischen unmittelbar angebundenen Prozessen und weiteren Prozessen, die beispielsweise eine Übertragung der Daten grundlegend gewährleisten, fließend ist.¹⁹ Die verborgenen Ebenen der digitalen Schrift sind mit der Umgebung verbunden, wie das Blog mit den verlinkten Webseiten. Der Hypertext, der auf dem Bildschirm zum Klicken einlädt, besteht auf der verborgenen Ebene der Schrift aus kooperierenden Programmen. Das literarische Schreiben der Autorin, das Erscheinen der Texte im Blog und das Lesen auf dem Bildschirm sind also flankiert von einer Vielzahl automatisierter Schreib- und Leseprozesse, in denen der literarische Text mit umfangreichen funktionalen Texten, das heißt Texten für die Steuerung von Prozessen oder Darstellungen, verbunden wird.

4. Archivobjekt und Rekonstruktion von Fassungen

Die Instabilität der Publikationsform, die *Ze zurrealism itzelf* durch Löschungen und Umschreibungen auch poetisch und performativ reflektiert, hat weitreichende Auswirkungen auf die Archivierung, die gerade durch diesen Charakter des Texts besonders wichtig wird. *Ze zurrealism itzelf* wurde vom Internet Archive seit 2010 mehrfach archiviert. Bei der Webarchivierung, wie sie das Internet Archive praktiziert

19 Zu Fragen der Abgrenzung von digitalen Objekten aus Erhaltungsperspektive siehe Dragan Espenschied: Digital Objecthood. In: Selçuk Artut/Osman Serhat Karaman/Cemal Yılmaz (Hg.): Technological Arts Preservation. Istanbul: Sabancı University 2021, <https://drupal.sakipsabancimuzesi.org/sites/default/files/2021-10/SSM_Technological_Arts_Preservation.pdf>, 116–139.

und maßgeblich weiterentwickelt, wird die gesamte Kommunikation zwischen Server und Browser aufgezeichnet. Gespeichert werden also nicht nur die im Browser gerenderten Seiten, sondern auch die Anfragen und weitere Interaktionen. Dieser Prozess ist im Fall des Internet Archives hochgradig automatisiert, kann aber auch manuell erfolgen. Aufgezeichnet wird dann der Weg durch eine Website. Das dadurch hergestellte Webarchiv kann mit einer entsprechenden Abspielumgebung so bereitgestellt werden, dass die im Archivierungsdurchgang besuchten Seiten mit den zugehörigen Ressourcen, die während der Archivierung mit gespeichert wurden, im Browser wieder aufgerufen werden können. Bei der Automatisierung des Vorgangs folgt ein sogenannter Crawler selbständig den Links auf einer Webseite, auch ohne manuelle Klicks. Es ist mit Blick auf den performativen Aspekt digitaler Schriftlichkeit und die zeitliche Instabilität des hier beschriebenen Blogs vielleicht bemerkenswert, dass eine der aktuell maßgeblichen Initiativen im Bereich Webarchivierung »Webrecorder« heißt.²⁰ Denn mit dem Begriff des Recordings oder der Aufzeichnung wird die Archivierung als zeitlich ausgedehnter Prozess beschrieben, die nicht nur Textinhalte, sondern auch technische Paratexte und Interaktionen archiviert. Dazu gehören der Seitenquelltext, zusätzliche Ressourcen wie etwa in externen Dateien gelagerte Informationen zur Darstellung der Seite (Schriftart, Schriftgröße usw.) sowie die zwischen Browser und Server ausgetauschten Parameter zur Steuerung der ausgelieferten Informationen, zum Beispiel Angaben zur voreingestellten Sprache oder Zeitzone.

Für das Blog *Ze zurrealism itzself*, das durch Löschungen und Umarbeitungen immer wieder im Gesamten verändert wurde, hängt die Struktur des archivierten Objekts also maßgeblich von der Archivierungsstrategie ab. Die Archivierungen von *Ze zurrealism itzself* im Internet Archive wurden dabei unregelmäßig durchgeführt, und die Seite wurde

20 Webrecorder Project, webrecorder.net, Homepage. Archiviert: <https://web.archive.org/web/2023112220949/https://webrecorder.net/> (zuletzt abgerufen am 22.11.2023).

in der Regel nicht vollständig archiviert. Beide Aspekte, Unregelmäßigkeit und Unvollständigkeit, sind durch die bereits beschriebene Publikationsform und die künstlerischen Möglichkeiten des Schreibens im Internet bedingt und hängen außerdem mit der Korpusbildung und der Archivierungsstrategie des Internet Archive zusammen. Vermutlich aufgrund von Redundanzvermeidung sind im Internet Archive keine vollständigen Fassungen für bestimmte Zeitpunkte archiviert. Oftmals gibt es für einen erkennbaren Zeitpunkt nur die erste Blogseite oder einen einzelnen Blogbeitrag, für andere Zeitpunkte wurden dagegen mehrere Blogbeiträge oder eine Übersichtsseite archiviert. Mit Blick auf eine einzelne Seite des Blogs finden sich im Internet Archive also unterschiedliche Fassungen, die im Interface mit einer Zeitleiste zeitlich geordnet dargestellt werden. Mit Blick auf das ganze Blog sind aber aufgrund der jeweils meist nur punktuellen Speicherung von Zuständen einzelner Seiten keine Fassungen nach Zeitschnitten verfügbar. Vielmehr werden im Archiv die einzelnen Seiten zu einem Gesamtobjekt zusammengeführt, die nicht notwendig aus einem eng abgegrenzten, quasi identischen Zeitraum stammen, sondern zum Teil aus unterschiedlichen Archivierungen zusammengeführt werden. Oder anders ausgedrückt: Ziel des Archivs ist es, für jeden Link auch eine Seite bereitzustellen. Dabei kann auch auf Seiten aus früheren oder späteren Archivierungen zurückgegriffen werden, sodass eine Bewegung durch das archivierte und rekonstruierte Blog auch mit einer Bewegung durch die Zeit verbunden sein kann. Die Lektüre eines zeitlich eingegrenzten Zustands des gesamten Blogs wird dadurch erschwert. Das Objekt ist darüber hinaus lückenhaft, denn die Archivierung ist nicht mit den Änderungen synchronisiert, sodass nicht ausgeschlossen werden kann, dass Änderungen der Autorin an einem Text nicht archiviert sind.

5. Fazit

Im vorliegenden Beitrag haben wir den transitorischen Charakter literarischer Blogs am Beispiel von *Ze zurrealism itzself* zunächst aus der Lesendenperspektive und sodann mit Blick auf die Textstruktur analysiert.

Die Ereignishaftigkeit flüchtiger Leseerlebnisse wird durch die digitale Publikation bzw. durch das Schreiben im Internet ermöglicht. Bezogen auf das digitale Schreiben und die Schrift im Internet unterscheiden wir verschiedene Ebenen. Der digitale Raum ermöglicht die Integration von Fotos in den Text, die auf anschauliche Weise das körperliche, handschriftliche Schreiben ausstellen und auf die besonders greif- und fühlbare Materialität der Haut und des Papiers aufmerksam machen. Die Bilder von Tinte auf der Haut am Hals oder von schriftförmigen Abdrücken auf der Oberschenkelhaut lenken die Aufmerksamkeit indirekt auch auf den digitalen Schriftträger, der die vorliegenden Texte im Blog *Ze zurrealism itzelf* deutlich dominiert. Das Bild von sechs kleinen, eng beschriebenen Post-its zeigt zudem die Begrenztheit der Ressource Papier oder auch einfach des praktischen Papierzuschnitts an, zu der die unbegrenzten Möglichkeiten im Digitalen im Gegensatz stehen. Die erste Ebene ausgestellter Handschriftlichkeit wird ergänzt um die Ebene der digitalen Schrift auf dem Bildschirm. Beide Ebenen bewegen sich an der sichtbaren Oberfläche. Darunter liegt der Quelltext als Basis der Schrift, hier sind literarischer Text und funktionaler Text miteinander verbunden. Darüber hinaus ist auch dieser Text codiert, und es braucht Software, also Code, um den codierten Text zu prozessieren. Diese Kette von Bedingungen führt schließlich zu den binären Ladungszuständen der Maschine, die im Blogtext als Nullen und Einsen wieder auftauchen, in einer reflexiven Benennung der eigenen digitalen Bedingungen.

Das Internet ermöglicht sowohl die Ereignishaftigkeit der Lektüre transitorischer Literatur als auch die Archivierung der Anteile, die im Verlauf durchgeführter Änderungen und Löschungen verloren gehen. Hervorgerufen werden die Leseereignisse des verschwundenen oder veränderten Textes in unserem Beispiel nach wie vor von einer Autorin, während es sich bei den Archivierungen um automatisierte Prozesse handelt, auf die die Autorin kaum Einfluss hat. Aufführungen, Performances und transitorische Literatur unterscheiden sich durch ihre Flüchtigkeit von Werken. Lessing kommt zu der Formulierung »Aber die Kunst des Schauspielens ist in ihren Werken transitorisch«, ²¹ die ausge-

21 Lessing: Kunst des Schauspielers, 234.

hend von der Bedeutung des bleibenden Kunstwerks paradox anmutet. In der Theatertheorie werden transitorische Aufführungen und wird auch die Schauspielkunst von den literarischen Werken unterschieden. *Ze surrealism itzelf* erhält durch die Instabilität der Texte und weil mit der Publikation der Texte kein finaler Zustand mehr verbunden ist ebenfalls einen transitorischen Charakter und weist der Archivierung damit eine konstitutive Funktion zu, weil hier vorübergehende Textzustände festgehalten werden und das Vorübergehende wenn nicht festhalten, so doch im Ansatz sichtbar machen.

Vernakulärer Code oder die Geister, die der Algorithmus rief – digitale Schriftlichkeit im Kontext sozialer Medienplattformen

Christian Schulz

Geht es um digitale Schriftlichkeit im Kontext sozialer Medienplattformen, so ist eine Analyse des »Codes« im Backend gar nicht unbedingt das Entscheidende, wie ich mit diesem Beitrag zeigen möchte. Vielmehr gilt es in Zeiten von zunehmend auf maschinellem Lernen basierender Feeds, wie sie mittlerweile kennzeichnend für alle großen kommerziellen sozialen Medienplattformen sind, ganz konkret nach den Ebenen der Verflechtung von digitaler Schrift von Nutzer:innen einerseits und Algorithmen andererseits zu fragen.

Beide Phänomenbereiche, digitale Schriftlichkeit auf der einen und Algorithmen auf der anderen Seite sind genuin als prozessual zu denken und dabei in der Nutzungspraxis permanent aufeinander bezogen. Diese Verflechtung wird sowohl in der konkreten Nutzung als auch analytisch im Rahmen von User:innen-Interfaces produktiv gemacht, wobei allerdings in zentralen Arbeiten der *Interface Studies* häufig auch eine reduktionistische Aufteilung in Front- und Backend mitschwingt, die gerade mit Blick auf die mit dem Organisationsprinzip des Affekts einhergehende Algorithmisierung sozialer Medienplattformen als nicht mehr zeitgemäß erscheint.¹

1 Vgl. für eine kompakte Darstellung der Organisationsprinzipien sozialer Medien, zu denen neben dem »Affekt«, auch die »Liste« und »Reziprozität« gezählt werden können, auch Christian Schulz: *Infrastrukturen der Anerkennung. Eine Theorie sozialer Medienplattformen*. Frankfurt a.M./New York: Campus 2023, 185–194.

Dementsprechend möchte der erste Teil dieses Beitrags zunächst das Interface als »Schwelle« definieren und so gleichzeitig den Begriff der Schnittstelle kritisch hinterfragen, wie darüber hinaus einen Konnex zur Paratextforschung von Gérard Genette herstellen.² Genette denkt nämlich ganz im Sinne von Walter Benjamin den Begriff der »Schwelle« nicht als dichotom mit einer Unterscheidung in eine Innen- und Außenseite, sondern als »Zone des Übergangs«³, die genuin auf den prozessualen Charakter des Begriffs verweist und sich insofern auch für eine Interface-Theorie fruchtbar machen lässt.

Ausgehend von diesem theoretischen Setting werden anschließend zwei Beispiele digitaler Schriftlichkeit in Form von sogenannten *Captions* auf der Plattform Instagram diskutiert. Mit diesen *Captions* sind hier die von den Nutzer:innen verfassten Begleittexte zu ihren Beiträgen oder Postings gemeint.⁴ Im ersten Beispiel, das sich einem via Selfie dokumentiertem Krankenhausaufenthalt einer Influencerin widmet, wird deutlich gemacht, wie anhand von Hashtags und gesetzten Links gezielt nicht nur die Syntax der *Captions*, sondern bewusst die für eine Sichtbarkeit auf der Plattform sorgenden Algorithmen adressiert werden. Im zweiten Beispiel, das sich anhand von einem mittlerweile gelöschten Selfie einer weiteren Nutzerin widmet, werden die Unsicherheiten thematisiert, die beim Ausbleiben solch einer durch digitale Schriftlichkeit adressierten Sichtbarkeit mitunter auf den Plan treten, wie die in diesem Zusammenhang analysierte *Caption* samt Anschlusskommunikationen zeigen wird. Schlussendlich werden davon

2 Gérard Genette: *Paratexte. Das Buch vom Beiwerk des Buches*. Frankfurt a.M.: Suhrkamp 2001.

3 Walter Benjamin: *Gesammelte Schriften*. Bd. V. Frankfurt a.M 1972–1989: Suhrkamp, 617f.

4 Auch Likes und Follower:innen-Counter lassen sich mit dem Paratext-Begriff nach Genette fassen. Im Folgenden soll aber die digitale Schriftlichkeit in Form der *Captions* im Mittelpunkt stehen. Vgl. für eine Ausdehnung des Begriffs auch Johannes Paßmann/Lisa Gerzen/Anne Helmond/Robert Jansma: Formular und digitaler Paratext. Geschichte des Facebook-Accountnamens. In: Peter Plesner/Niels Werber/Burkhardt Wolf (Hg.): *Das Formular*. Metzler: Stuttgart 2022, 307–324.

ausgehend theoretische Überlegungen angestellt zu einem Überschuss des Imaginären, der auf beiden Seiten des Interface in der permanenten Verfertigung von digitaler Schriftlichkeit und Algorithmen erzeugt wird und sich damit nicht weniger als konstitutiv für das Funktionieren sozialer Medienplattformen erweist.

1. Das Interface als Schwelle

Voraussetzung für sämtliche Formen der Interaktion sowohl in sozialen Medien als auch für digitale Schriftlichkeit generell sind Interfaces. Mit dem Begriff des *Interface* wird hierbei eine Schnittstelle bezeichnet, i.d.R. zwischen Mensch und Maschine, die die technischen Verschaltungen dieser beiden Entitäten regelt und (un)sichtbar macht. Häufig wird der Begriff jedoch auf das *Graphical User Interface* (GUI), also die für alle Nutzer:innen sichtbare Seite am Frontend, reduziert.⁵ Dabei weist der Begriff des Interface an sich bereits darüber hinaus.⁶ So bezeichnet er neben dem Zugang von Menschen zur Maschine über die sichtbaren Benutzer:innenoberflächen hinaus auch die Vermittlung zwischen verschiedenen Programmen bzw. Programmteilen oder Hardwarekomponenten. Trotz dieser in erster Linie vermittelnden Funktion sind Interfaces aber auch als ein Effekt der Abschirmung zu begreifen, die aus User:innen-Perspektive den direkten Kontakt mit der technischen Entität verhindern (obwohl sie ihn gleichsam suggerieren).⁷ Auch Programmteile oder Bibliotheken werden so durch Interfaces nicht alleinig vermittelt, sondern durch jene als abgekapselte

5 Johanna Drucker: Humanities Approaches to Interface Theory. In: *Culture Machine* 12 (2011), 1–20; vgl. für einen Überblick zur Interface-Forschung auch Florian Hadler: Beyond UX. In: Ders./Alice Soiné/Danile Irrgang (Hg.): *Interface Critique* Vol. 1 2018.

6 Dies gilt zumindest für eine medienwissenschaftliche Perspektive.

7 Stefan Heidenreich: Icons. Bilder für User und Idioten. In: Birgit Richard/Robert Klanren/Stefan Heidenreich (Hg.): *Icons. Localizer 1.3*. Berlin: Die Gestalten Verlag 1998, 82–85; Florian Cramer/Matthew Fuller: Interface. In: Matthew Fuller (Hg.): *Software Studies. A Lexicon*. MIT Press: Cambridge London 2006, 149–152.

Instanz nutzbar. Insofern ist es nur folgerichtig, das Interface in einem relationalen und prozessualen Sinn zu denken, in dem Zugang und Abschirmung verhandelt werden. Allerdings gilt es hierfür, den Begriff der »Schnittstelle« kritisch zu hinterfragen. Denn dieser scheint eben doch die Grenzen zu vermittelnder Entitäten hervorzuheben, die oft in Form einer Dichotomie konzeptualisiert sind. Damit wird eine Differenz produziert, die spätestens in Anbetracht algorithmisierter Feeds von sozialen Medienplattformen zu kurz greift, weil die relevanten Prozesse nicht hinter dem ›Interface‹ ablaufen, sondern in einem permanenten Zusammenspiel zwischen User:innen, dem Interface selbst und den Algorithmen auf Seiten der Plattformen.

Hierfür möchte ich im Folgenden den Begriff der ›Schwelle‹ aufgreifen, wie er an anderer Stelle bereits vorgeschlagen wurde.⁸ In der Interface-Forschung ist der Begriff der ›Schwelle‹ durchaus präsent, wird aber fast immer synonym mit dem Begriff der ›Schnittstelle‹ verwendet. So thematisieren beispielsweise sowohl Alexander Galloway als auch Branden Hookway in ihren 2012 und 2014 erschienenen und für die Interface-Forschung einflussreichen Büchern den Begriff, ohne diesen jedoch systematisch zu verwenden.⁹ Hookway wie Galloway denken die Schwelle synonym zum Begriff der Schnittstelle und gehen von einer Innen-Außen-Dichotomie von Front- und Backend aus (also binären Zuständen bzw. Interfaces die weitere Interfaces beinhalten).¹⁰ Dabei führt Galloway mit dem ›Intraface‹ bereits einen interessanten Begriff ein, der zwar einer Innen-Außen-Dichotomie verhaftet bleibt – weil er lediglich eine feingliedrige Perspektive vorschlägt, in der das Interface eben mehrere Schichten oder Ebenen hat –, es aber in letzter Instanz

8 Christian Schulz/Tobias Matzner: Feed the Interface. Social-Media-Feeds als Schwellen. In: *Navigationen. Zeitschrift für Medien- und Kulturwissenschaften* 02/2020, 147–164.

9 Alexander R. Galloway: *The Interface Effect*. Cambridge/Malden: Polity Press 2012; Branden Hookway: *Interface*. Cambridge: MIT Press 2014.

10 Siehe hierzu insbesondere die Textstellen im Vorwort von Galloway: *The Interface Effect*, vii sowie Hookway: *Interface*, 5.

immer ein Außen gibt, das dem Inneren entgegengestellt wird.¹¹ Dennoch ist damit bereits das Potenzial gegeben, solche Dichotomien zu umgehen, womit sich Galloway interessanterweise ganz nah an dem bewegt, was im agentiellen Realismus als »Intraaktion« bezeichnet wird.¹²

Karen Barads Ausführungen zur Intraaktion wiederum sind trotz oder gerade wegen der radikal prozessualen Perspektive allerdings auch sehr von Fragen der Verschränkung von Ontologie und Epistemologie in wissenschaftlichen Praktiken beeinflusst. Insofern steht die gegenseitige Konstitution von Beobachtendem und Beobachtetem im Vordergrund. Die spezifische Dynamik von algorithmisierten sozialen Medienplattformen lässt sich damit jedoch nicht fassen. Eine von Barads Intraaktionen inspirierte Perspektive auf Interfaces, welche den Begriff der »Schwelle« aufgreift, diesen aber zugleich einer notwendigen Schärfung unterzieht, bietet hier einen analytischen Vorteil.

Wie Bernhard Waldenfels schreibt, ist die Schwelle »ein Ort des Übergangs, ein Niemandsort [...], den man hinter sich lässt, aber nie ganz.«¹³ Als ein solcher »Ort der Schweben«¹⁴ ist sie jedoch klar von der Grenze und damit der Schnittstelle zu trennen, wie bereits Walter Benjamin in seinem Passagen-Werk schreibt:

»Schwelle und Grenze (und damit die Schnittstelle; Anm. d. Verf.) sind schärfstens zu unterscheiden. Die Schwelle ist eine Zone. Und zwar eine Zone des Übergangs. Wandel, Übergang, Fliehen liegen im Worte

11 Alexander R. Galloway: *Außer Betrieb. Das müßige Interface*. Köln: Walther König 2010, 27. Siehe ausführlicher zu dieser Dichotomie auch Schulz/Matzner: *Feed the Interface*, 148–150.

12 Karen Barad: *Agentieller Realismus. Über die Bedeutung materiell-diskursiver Praktiken*. Berlin: Suhrkamp 2012, 20. Dort heißt es: »[...] die Relata existieren nicht schon vor den Relationen; vielmehr entstehen Relata-in-Phänomenen durch spezifische Intraaktionen.«

13 Bernhard Waldenfels: *Sinnesschwellen. Studien zur Phänomenologie des Fremden 3*. Frankfurt a.M.: Suhrkamp 1999, 9.

14 Ebd.

schwellen und diese Bedeutungen hat die Etymologie nicht zu übergehen.«¹⁵

Dementsprechend lässt sich keine Demarkationslinie zwischen Innen und Außen ziehen, sondern es handelt sich um einen Bereich des Übergangs und der Transformation, weshalb sich in der Schwelle konstante Verschiebungen vom einen zum anderen Zustand ereignen, ohne dass diese klar fixierbar wären. Gleichzeitig bleibt in einem so gedachten Konzept der ›Schwelle‹ aber die Existenz eines unbestimmten oder ambivalenten Raums relevant, während hingegen für Barad nur eine Intraaktion (d.h. ein Paar Beobachtetes/Beobachtendes) in die nächste (das nächste Paar) übergehen kann. Der Begriff der ›Schwelle‹ nach Benjamin erkennt diesen unbestimmten oder ambivalenten Raum innerhalb der Intraaktion aber an.

Und in genau diesem unbestimmten Raum sind Interfaces samt der für die User:innen zugänglichen grafischen Benutzer:innenoberflächen zu verorten, in die sie ihre Eingaben tätigen können. Für eine digitale Schriftlichkeit ist besonders bemerkenswert, dass etwa Gérard Genette in seinem Standardwerk zur Paratextforschung, dessen Originaltitel bezeichnenderweise *Seuils* (frz. »Schwellen«) lautet, bereits in der Einleitung den Paratext als »unbestimmte Zone« und insofern als eine eben solche Schwelle definiert, ohne eine wie auch immer geartete Dichotomie von Innen und Außen zu reproduzieren, womit er explizit an Benjamin anknüpft.¹⁶

Im Folgenden möchte ich nun exemplarisch zwei Beispiele von solchen Paratexten im Kontext sozialer Medienplattformen herausgreifen, die zeigen, dass die Nutzer:innen ganz konkret mithilfe der oben vorgestellten Captions versuchen, Einfluss auf den Algorithmus

15 Benjamin: *Gesammelte Schriften*, Bd. V, 617f.

16 Genette: *Paratexte*, 10; Vgl. auch Georg Stanitzek: Texte, Paratexte, in Medien: Einleitung. In: Ders./Klaus Kreimeier (Hg.): *Paratexte in Literatur, Film, Fernsehen*. Berlin: Akademie Verlag 2004, 3–19, hier 6; Auch Alexander Galloway stellt in seinem Buch bereits den Bezug zu Genette her, allerdings unter Prämisse der Dichotomie von Innen und Außen (vgl. Galloway: *The Interface Effect*, 42).

auszuüben bzw. Unsicherheiten darüber kommunizieren bei einer ausbleibenden Sichtbarkeit ihrer Beiträge. In diesem Sinne sind sowohl die Kuratierung der Social-Media-Feeds als auch die daran geknüpften Vorstellungswelten oder *Imaginaries* der Nutzer:innen über die Funktionsweise der Algorithmen gewissermaßen Teil eines ›vernakulären Codes‹,¹⁷ mit dem die Nutzer:innen während der alltäglichen Nutzung den Algorithmus durch ihre Eingaben ›programmieren‹, was mal mehr, mal weniger gut gelingt und mitunter zu Verunsicherungen führen kann.

2. Digitale Schriftlichkeit I: Hashtags

Das erste Beispiel ist eine Caption unter einem ganz klassisch aufgenommenen Selfie vom 28. November 2017, das die Agenturinhaberin und Influencerin @travala im Bett liegend mit einer Gesichtsmaske und offensichtlich verweinten Augen zeigt. Unterlegt ist dieses Posting mit einem Zitat der Band *Faithless* aus dem Song »Insomnia«.

Hier heißt es im ersten Teil der Caption den Songtext zitierend: »This insomniac, take an original tack. Keep the beast in my nature under ceaseless attack, I need to sleep, I can't get no sleep«.

Dieser Teil der paratextuellen Einlassungen verweist zunächst auf offensichtlich vorhandene Schlafstörungen, die die Nutzerin während sie das Selfie hochgeladen hat, zu plagen scheinen, als auch die Infektion wegen derer sie im Krankenhaus ist.¹⁸

17 Ich beziehe mich bei dem Begriff des ›Vernakulären‹ auf die Anthropologin Margaret Lantis, die damit eine Alltagssprache und das Wissen von Nicht-Expert:innen hervorhebt und in den Vordergrund rückt bzw. dieses Wissen im Vergleich zu einem wie auch immer gearteten Expert:innenwissen aufwertet. Vgl. Margaret Lantis: Vernacular Culture. In: *American Anthropologist* 62(2), 1960, 202–216.

18 @travala befindet sich aufgrund einer Infektion mit einem Keim im Krankenhaus, den sie sich bei einem Routineeingriff ein paar Tage zuvor eingefangen hat.

Im zweiten Teil der Caption heißt es dann allerdings ziemlich überraschend:

»But i can find an #overnight #beauty #mask (to #apply without a #mirror #nomirrorsince16days #selfie #application) @royalfernskincare to get #miracle #morning #glow and #deep #skin #hydration #sleepless but #plumpy #hospital #night #life #selfcare #solutions«

Unübersehbar fällt hier neben den obligaten inhaltlichen Verweisen auf die eigene Befindlichkeit und die Umstände strukturell zunächst das Verschwimmen der regulär syntaktischen Textbausteine (»But i can find an«; »to«; »without a«; »to get«; »and«) mit den vielen Hashtags auf. Wobei die Caption bereits nach dem fünften Wort fast ausschließlich aus Hashtags besteht, die fortan und obgleich etwas hölzern anmutend, doch einer Art Syntax folgen.

Es werden, insbesondere mit den Schlagworten beziehungsweise den Tags »#selfcare« und »#solutions«, gleichermaßen vermeintliche Ansätze zur Lösung aus der Situation aufgezeigt wie auf extrem populäre Hashtags zurückgegriffen.

Bemerkenswert ist nun, dass das Zentrum des Satzes eine Verlinkung zu einem Unternehmen enthält, das Hautpflege-Produkte herstellt. Dieses Unternehmen (@royalfernskincare), dessen Gesichtsmaske @travala wahrscheinlich aufgetragen hat, ist auch im Foto getaggt und wenn man die Hashtags »#apply« und »#application« samt der syntaktischen Versatzstücke betrachtet, handelt es sich hier wohl um einen Post mit dem sich @travala als Werbeträgerin für die Produkte von *Royal Fern* zu imaginieren scheint, weshalb sie auch explizit von »application« spricht.

An dieser Komposition zwischen einem Selfie mit kosmetischer Gesichtsmaske und einer gewissermaßen etwas widersprüchlichen Caption, die einerseits aus einem Song-Zitat der Band *Faithless* besteht, in dem die persönliche Krisensituation aufgegriffen wird, und die sie andererseits als (imaginierte) Werbeträgerin für ein Unternehmen präsentiert, das Hautpflegeprodukte herstellt, lässt sich die Rolle von Hashtags

im Kontext sozialer Medienplattformen auf mehreren Ebenen analysieren:

Zunächst ist auf einer diskursiv-formalen Ebene zu konstatieren, dass der Beitrag in Form des Selfies erst einmal ein sehr stereotyper Content ist, was dem von Uwe Pörksen geprägten Begriff des ›Visiotyps‹ entspricht. Pörksen bezeichnet damit bereits in den 1990er Jahren durch die Entwicklung der Informationstechnologien begünstigte Typen von sich rasch standardisierender Visualisierung, die zu bestimmten Zeitpunkten den vorherrschenden Zugriff auf ›die Wirklichkeit‹ darstellen.¹⁹ Beim Visiotyp des Selfies war dies insbesondere in den Jahren 2016 und 2017 unverkennbar und äußerte sich u.a. darin, dass eine Zeit lang alle möglichen Fotos, die auf sozialen Medienplattformen eingestellt wurden, nicht nur in der breiten Öffentlichkeit als Selfies beschrieben wurden.

Auf einer fototheoretischen Ebene ließe sich mit Roland Barthes nun auch ein *punctum*, mit dem die Betrachter:innen affizierenden, verweinten Augen ausmachen. Barthes bezeichnet mit dem Begriff des *punctum* ein kleines, unscheinbares fotografisches Detail, das die Aufmerksamkeit der das Foto Betrachtenden auf sich zieht. Als ein solches kann es der bewussten Intention der Fotograf:innen zuwiderlaufen und sich quasi subversiv im intendierten Bildinhalt einnisten.²⁰

Beide Lesarten sind für ein Verständnis auf visueller Ebene wichtig und im Sinne des vorgeschlagenen Ansatzes eines Zusammendenkens von Innen und Außen bzw. Front- und Backend des Interfaces selbstverständlich miteinzubeziehen.

Allerdings gilt es auch, den Blick über die sichtbare Ebene (oder das Frontend) hinaus zu erweitern, weshalb das Selfie nicht in erster Linie als Bild, sondern vielmehr als (affektive) Geste und damit als der Punkt, an dem sich der Körper in die medialen Infrastrukturen einschreibt, ver-

19 Uwe Pörksen: *Weltmarkt der Bilder. Eine Philosophie der Visiotype*. Stuttgart: Klett-Cotta 1997, 27.

20 Roland Barthes: *Die helle Kammer. Bemerkungen zur Photographie*. Frankfurt a.M.: Suhrkamp 1980, 60.

standen werden soll.²¹ Als Komplementärfigur der Schwelle kennt die Geste ebenfalls kein Innen und Außen und ist darüber hinaus auch als basale Bedingung von Kommunikation und deren Realität zu bezeichnen.²² Denn über die Geste entsteht zuallererst ein sozialer Kontakt, der vermittelt über das affizierende Element allerdings von der Gegenseite oder der jeweiligen Peer-Group aufgegriffen werden muss, um nicht zu versanden und sich nicht zuletzt aufgrund dieser Unbestimmtheit, die ständig die Möglichkeit des Scheiterns hinsichtlich von Anschlusskommunikationen impliziert, immer auch des Imaginären in Form von Perspektivübernahmen bedienen muss.²³

Zu einer solch affektiven Geste wird das Selfie von @travala nun aber erst durch die in es verwobenen Hashtags in der Caption, die auf inhaltlicher (und vielleicht persönlicher) Ebene nicht nur ebendiese Spannung zwischen der eigenen Befindlichkeit und imaginerter Werbeträgerin widerspiegeln. Auf einer infrastrukturellen Ebene markieren die Hashtags und die Verlinkung also das Vehikel, das das Selfie die Schwelle des Interfaces überhaupt erst überschreiten und es somit auch zum Zirkulieren bringen lässt.

Denn die Hashtags stellen ein wesentliches Element für die Sichtbarkeit auf der Plattform dar und sagen dem Algorithmus etwa, dass bestimmte Wörter präferiert und andere vernachlässigt werden sollen (je nach individuellem Kontext). So werden Beiträge nach bestimmten Clustern sortiert, die nicht nur an individuelle Suchanfragen von Nutzer:innen, sondern auch an die Filter-Mechanismen sozialer

21 Vgl. zum Selfie als Geste auch Paul Frosh: *The Poetics of Digital Media*. Cambridge/Malden: Polity Press 2019, 115–137.

22 Reinhold Göring: Einleitung. In: Ders./Timo Skrandies/Stephan Trinkaus (Hg.): *Geste. Bewegungen zwischen Film und Tanz*. Bielefeld: transcript 2009, 9–18, hier 10f.

23 George Herbert Mead bezeichnet diese Perspektivübernahmen mit der Figur des ›generalisierten Anderen‹. Dieser ›generalisierte Andere‹ sei die Haltung, und damit Normativität, der ganzen Gesellschaft und wesentlich für die Aushandlung von Sozialität (George Herbert Mead: *Geist, Identität und Gesellschaft*. Frankfurt a.M.: Suhrkamp 1973, 196 und 218).

Medienplattformen rückgebunden sind und die nahezu bei allen Plattformen aus einer Kombination von *kollaborativem* und *inhaltsbasiertem Filtern* bestehen²⁴, die zudem oft noch mit verschiedenen Verfahren des maschinellen Lernens gekoppelt werden (z.B. Random Forest-Entscheidungsbäumen). Insofern reihen sich die Captions als paratextuelle Einlassungen gepaart mit einem auf der Bildebene affizierenden Inhalt also ein in die Riege von weiteren Plattformeaktoren wie Likes oder Kommentaren. Das Zusammenwirken eines auf Bildebene affizierenden Inhalts und den paratextuellen Einlassungen, zu denen die Captions samt der Hashtags ebenso gehören wie die Likes und Kommentare der Follower:innen, adressiert also gewissermaßen die Zwischenräume dieser sozio-technischen Verschaltungen, die mit der Denkfigur des Interfaces als Schwelle beschrieben werden können. Im vorliegenden Fall adressiert die Nutzerin also bewusst diese Schwelle, indem nahezu ein vollständiger Satz in Form von aneinandergereihten und mitunter extrem populären Hashtags (z.B. #selfie, #selfcare oder #beauty) für die »Programmierung« des Algorithmus im Hinblick auf die eigene Sichtbarkeit des Beitrags auf der Plattform sorgen soll.

Digitale Schrift im Kontext der Plattform ist hier durch den prozessualen Charakter also explizit als digitale Schriftlichkeit zu denken, die in Form der Caption, und auch je länger sie ausfällt, nicht nur für eine längere Verweildauer von Nutzer:innen auf dem Posting sorgt (was ein positives Signal für den Algorithmus darstellt). Vielmehr stellt dieser vernakuläre Code überhaupt erst die Voraussetzung für eine Zirkulation auf der Plattform dar. Insofern ist die Caption samt Hashtags zwar als intentionale Handlung der Nutzer:innen zu charakterisieren, die aber, sobald sie die Schwelle des Interface überquert, sich prozessual verflüssigt und mitunter Irritationen hervorrufen kann, wie im Folgenden das zweite Beispiel zeigt.

24 *Kollaboratives Filtern* basiert auf Mustern oder Clustern im Nutzungsverhalten aller User:innen einer Plattform (Nutzer:innen, die X mochten, mögen auch Y), während *inhaltsbasiertes Filtern* um die jeweiligen Inhalte herum organisiert wird und im Kontext sozialer Medienplattformen oft mit Popularitätsmarkern wie Likes und Views verknüpft wird.

3. Digitale Schriftlichkeit II: Unsichtbarkeit

Exemplarisch für so eine Irritation steht eine Instagram-Diskussion, die ich im Oktober 2017 auf dem Account einer anderen Nutzerin beobachten konnte.

Diese postete am 7. Oktober 2017 ein später gelöscht Selfie. Versehen wird das Posting mit folgender Caption und 23 für die Nutzerin recht gewöhnlichen Hashtags wie z.B. #selfie oder #instamum:

»Ich bin ein wenig ratlos warum eure Kommentare & Likes im [sic] letzter Zeit so zurückgegangen sind. Seht ihr meine Posts überhaupt noch? [...] Habt ihr ähnliche Probleme?«

Was zunächst wie eine persönliche Adressierung der Follower:innen über mangelndes Engagement vonseiten ebenjener anmutet, erweist sich scheinbar als direkte Reaktion auf ausbleibende Likes und Kommentare wie auch der nachfolgende Kommentar einer Followerin zeigt:

»Mir geht es genauso. Bin auch sehr traurig. Ich habe das Gefühl Insta [sic!] hat wieder etwas geändert. Ich sehe deine Fotos noch, aber zeitlich mega versetzt und nicht immer, habe ich das Gefühl«

Das Gefühl, dass die Plattform »irgendetwas« geändert haben muss, ist also ubiquitär, allerdings zu gleichen Teilen auch merkwürdig diffus, wie weitere Kommentare beziehungsweise Unterhaltungen unter dem geposteten Selfie verraten: »*Würde mich halt interessieren, woran es liegt*« heißt es da von der Nutzerin in Response auf die Aussage einer anderen Followerin, dass das ganz komisch sei und sie an manchen Tagen das Gefühl habe, keinerlei Posts mehr von ihr zu sehen. Und generell, so eine weitere Followerin, sei das ja ungerecht, da man sich manchmal so viel Mühe für ein Foto gebe, das dann so wenig Likes bekomme.

Eine vierte Followerin bringt es schließlich auf den Punkt: »*Das ist der Algorithmus! Der macht manchmal einfach was er will.*« Dieser letzte Kommentar suggeriert gewissermaßen eine Art zeitweisen Kontrollverlust über »den« Algorithmus, mit dem es eben manchmal einfach durchgeht

und der dann schlichtweg macht, was er will, womit letztlich nicht nur eine Vermenschlichung algorithmischer Systeme stattfindet.

Interessant ist hier vielmehr, dass mit dem Framing dieses Satzes Kontrolle über ›den‹ Algorithmus als Normalzustand vorausgesetzt wird und eben erst durch dieses Störmoment in Form dessen, dass die jeweiligen Beiträge nicht mehr bei den Follower:innen ankommen, Agentialität an die Technik delegiert wird. Die Vermenschlichungsgeste in den Vorstellungswelten der Nutzer:innen mit der Zuschreibung eines autonomen Willens (›der macht manchmal einfach was er will‹) an die Technik, dient hier also als simpelste Form einer Erklärung für die von der Nutzerin beklagten Sichtbarkeitsprobleme, womit gewissermaßen eine Subjektivierung des Algorithmischen auf Seiten der Nutzer:innen angenommen wird.

Das zweite Beispiel verweist somit auf die Störmomente jenes vernakulären Codes, der im ersten Beispiel noch via paratextueller Captions von der Nutzerin explizit über die Ebene populärer Hashtags wie »#selfie«, »#selfcare« oder »#beauty« in der Syntax bespielt wird. Einerseits wird dadurch deutlich, dass der vernakuläre Code von Nutzer:innen immer auch an die jeweiligen Zirkulationsbedingungen der Plattformen gebunden ist, worin gewissermaßen auch der hierarchische Anteil besteht.²⁵ Andererseits wird damit jedoch auch zwangsläufig eine Art imaginärer Überschuss erzeugt, der insbesondere in Störmomenten sichtbar wird, wie das zweite Beispiel zeigt

25 Ariella Azoulay thematisiert eben diesen hierarchischen Aspekt des Vernakulären im Zusammenhang mit den kolonial-institutionalisierten Zirkulationsbedingungen von Fotografien. Darüber hinaus verweist bereits die Begriffsgeschichte des Wortes (von lat. *verna* für Haussklave) an sich auf die eingeschriebenen hierarchischen Aspekte im Begriff des ›Vernakulären‹. Vgl. auch Ariella Azoulay: *Market Transactions Cannot Abolish Decades of Plunder*. In: Tina Campt/Brian Wallis/Marianne Hirsch/Gil Hochberg (Hg.): *Imagining Everyday Life: Engagements with Vernacular Photography*. Göttingen: Steidl 2020, 47–60, hier 59. Mit Dank an Vera Knippschild für diesen Hinweis.

hat, ohne deshalb je einen Blick in die Black Box des Algorithmus im vermeintlichen Backend gewähren zu können.²⁶

Über eine solche von den Nutzer:innen ausgehende Perspektive über die (vermeintliche) Funktionsweise des Algorithmus wird auch deutlich, dass das Imaginäre der Nutzer:innen hier nicht einfach im Sinne einer ideologischen Verblendung der Subjekte oder gar der ganzen Software als Ideologie, wie dies etwa Wendy Chun formuliert hat und worauf sich auch Galloway in seinem Interface-Buch stützt,²⁷ beschrieben werden kann. Denn die mit dem algorithmischen Ranking verknüpften Captions der Nutzer:innen samt Hashtags, werden ähnlich wie die Anschlusskommunikationen im zweiten Beispiel (hier über die Kommentare) gewissermaßen direkt Teil der Infrastruktur sozialer Medienplattformen, weshalb hier auch von einer Art mehr oder weniger im- und expliziten ›Programmierung‹ durch die Nutzer:innen gesprochen werden kann. Der Begriff des ›vernakulären Codes‹ soll genau dies verdeutlichen und den Anteil des vernakulären Wissens alltäglicher Nutzer:innen hervorheben, gleichzeitig aber auch die Eingebundenheit in die hierarchischen Strukturen der Plattformen betonen.

Insofern ist es hier auch zu kurz gegriffen, von einer kybernetischen Einschleifung der User:innen in algorithmische Datenverarbeitung auszugehen, denn ebenso wenig, wie sich Technisches und Imaginäres trennen lassen,²⁸ sind auch digitale Schriftlichkeit und Algorithmen untrennbar durch jenen vernakulären Code aufeinander bezogen, der in der alltäglichen Nutzung im- und explizit erzeugt wird und über die Schwelle des Interface die Algorithmen gewissermaßen (mit)programmiert.

26 Taina Bucher: The algorithmic imaginary: exploring the ordinary affects of Facebook algorithms. In: *Information, Communication & Society* 20 (2017), 30–44.

27 Wendy Hui Kyong Chun: Über Software, oder: Die Beharrlichkeit visuellen Wissens. In: Kathrin Peters/Andrea Seier (Hg.): *Gender & Medien Reader*. Diaphanes: Zürich/Berlin 2016, 279–302; Galloway: *The Interface Effect*, 70.

28 Christian Schulz: A New Algorithmic Imaginary. In: *Media, Culture & Society* 45/3 (2023), 646–655.

Dieser vernakuläre Code adressiert damit nicht die ›reale‹ oder ›richtige‹ technische Funktionsweise des Algorithmus im Backend, sondern speist sich maßgeblich aus dem Imaginären der Nutzer:innen. Dies umfasst neben der Antizipation der technischen Funktionsweise von Algorithmen oder normativen Implikationen, wie sie z.B. Figuren des »verallgemeinerten Anderen«²⁹ adressieren und konstitutiv für die Aushandlung von Sozialität sind, auch die ökonomischen Interessen der Plattformen, die beispielsweise einflussreiche Nutzer:innen (meist Influencer:innen) über die Funktionsweise bestimmter Algorithmen informieren, ohne freilich technisch allzu sehr ins Detail gehen zu können.³⁰

Überhaupt geht es auf der anderen Seite des Interface, also auf Seiten der Plattformen, sowohl darum, ein möglichst genaues Datenabbild mit diesem vernakulären Code der Nutzer:innen zu erstellen als auch mit den von den Nutzer:innen annotierten Fotos künstliche Intelligenz zu trainieren, weshalb es neben Tests mit Funktionen, die etwa Fotos automatisch bestimmte Hashtags zuordnen auch großangelegte Trainingsszenarien gibt, in denen die von Nutzer:innen annotierten Fotos dazu verwendet werden, automatisierte Bilderkennung zu verbessern.³¹ Dieser Aspekt hat sich in Zeiten von generativer KI und OpenAIs *Dall-E 2* noch einmal zugespitzt, denn auch dort werden durch Social-Media-

29 Mead: *Geist, Identität und Gesellschaft*, 196.

30 Vgl. etwa Teri Gevinson: Who I be Without Instagram? An Investigation. In: *The Cut* 16. September 2019, online: <https://www.thecut.com/2019/09/who-would-tavi-gevinson-be-without-instagram.html> (zuletzt abgerufen am 30.06.2023); Minna Ruckenstein: *The Feel of Algorithms*. Oakland: University of California Press 2023, 69–97.

31 Lucas Matney: Facebook is using your Instagram photos to train its image recognition AI. In: *Techcrunch*, 2. Mai 2018, online: <https://techcrunch.com/2018/05/02/facebook-is-using-your-instagram-photos-to-train-its-image-recognition-ai/> (zuletzt abgerufen am 30.06.2023).

Nutzer:innen mit Hashtags annotierte Fotos zum Training für die dahinterliegenden KI-Modelle genutzt.³²

4. Fazit

Abschließend und zusammenfassend können auf einer theoretischen Ebene zwei Thesen zum Stellenwert digitaler Schriftlichkeit im Kontext sozialer Medienplattformen formuliert werden:

1. Digitale Schriftlichkeit ist immer an Interfaces gekoppelt und als solche maßgeblich an der prozessualen Verfertigung von Nutzer:innen und Algorithmen beteiligt. Wichtig zu betonen ist hierbei, dass diese Entitäten, Nutzer:innen und Algorithmen, nicht im Vorhinein feststehen bzw. vollständig determiniert sind, sondern sich gewissermaßen intraaktiv herauschälen, wobei digitaler Schrift ein wesentlicher Stellenwert durch Captions, Hashtags und Kommentarfunktionen zukommt. Gerade deshalb ist es wichtig, das Interface als Schwelle zu denken und weniger als Schnittstelle, in der bereits vorgefertigte Entitäten verschaltet werden.
2. Darauf aufbauend lässt sich die permanente Aufeinanderbezogenheit von Nutzer:innen und Algorithmen nur durch die Vorstellungswelten der Nutzer:innen adäquat erklären, deren Vehikel oder Medium wiederum eine in den Interfaces der Plattformen verankerte Möglichkeit zur Eingabe dieser Imaginaries ist. Diese sind somit wesentlich an digitale Schriftlichkeit gebunden. Die in den Beispielen thematisierten Captions u. a. bestehend aus Hashtags, aber auch die Anschlusskommunikationen oder verkürzten Funktionen wie Likes (originär einmal als Verkürzung für Zustimmung eingeführt), lassen sich als initiale oder affizierende Momente lesen, die die Zirkulation

32 Andreas Ervik: Generative AI and the Collective Imaginary: The Technology-Guided Social Imagination in AI-Imaginesis. In: *The Interdisciplinary Journal of Image Sciences* 37/1 (2023), 42–57, hier 49.

auf der Plattform erst in Gang bringen und somit zur permanenten Verfertigung von Nutzer:innen und Algorithmen beitragen.

Insofern kann die Rolle, die digitale Schriftlichkeit sowohl bei der Aushandlung von Sozialität über die Plattformen als auch für deren technische Funktionsweise spielt, kaum überschätzt werden.³³ Der vernakuläre Code, der also einen imaginären Überschuss aufseiten der Nutzer:innen erzeugt, ist somit auch mitnichten als eine Einbahnstraße zu bezeichnen, die algorithmisch determinierend ist, denn es geht schließlich auch oder vielmehr immer noch um die Aushandlung von Sozialität. Dennoch ist der Geist, den der vernakuläre Code atmet, algorithmischer Natur, obgleich er wie ausgeführt Schwellencharakter hat.

33 Hier wäre sicher mit Blick auf Plattformen wie Snapchat oder TikTok noch einmal zu differenzieren. Gleichwohl sind insbesondere auf TikTok Hashtags, und damit digitale Schrift, ebenfalls ein wesentlicher Teil der Zirkulation von Clips. Zudem ist es in der Forschung zu sozialen Medien zunehmend wichtiger, den plattformübergreifenden Charakter nicht nur von Plattformfunktionen wie algorithmisierten Feeds oder Like-Buttons zu berücksichtigen, sondern auch von Inhalten, die von Plattform zu Plattform wandern. Und hier spielt digitale Schriftlichkeit in Form von Hashtags eine Schlüsselrolle.

Humanities of the Digital

Philologische Perspektiven auf Source Codes als Beitrag einer computerarchäologischen Knowledge Preservation

Stefan Höltgen

0. Einleitung

Der digitale Wandel hat alle Bereiche des Wissens erfasst. Informationen kursieren als Daten in Netzwerken und stehen an beinahe jedem Ort der Welt auf Computern zur Verfügung. Klassische Medieninhalte haben sich von ihren materiellen Substraten gelöst und werden als Dateien oder Datenstreams auf mobilen Endgeräten oder computerisierten Großbildschirmen wiedergegeben. Die Software selbst, die solche Geräte in Form von Apps oder Programmen zur Konversion des jeweiligen Mediums im Digitalcomputer benötigt, ist als Sourcecode in algorithmisch sorgsam edierten Repositories abgelegt. Papier, so scheint es, wird langsam zu einem medientechnischen Relikt und Schrift residiert häufig nur noch als grafischer Gestalteffekt auf Displays.

Nicht trotz, sondern wegen dieser eskalierenden Digitalisierung und Virtualisierung unserer Symbolwelten wächst das Schrifttum in geometrischem Maße, denn »[d]er Computer ist eine Technologie der symbolischen Repräsentation und der Kommunikation, kurz – eine

Technologie des Schreibens.«¹ Mit ihm wird immer mehr Text geschrieben, weil auf immer mehr digitalen Kanälen produziert und rezipiert werden kann, und weil diese Kanäle zur Hälfte² selbst aus Schrift gebaut sind: aus Software – das sind Programmcodes und Daten, die unsichtbar in die Unterflächen der Digitaltechnologie geschrieben dafür sorgen, dass auf den Oberflächen etwas lesbar wird.³ Die Bedeutung von Schrift im ›Zeitalter der Digitalisierung‹ kann also kaum unterschätzt werden, treibt sie doch all unsere symbolverarbeitenden Maschinen an und bringt damit *en passant* eine eigene, arkane Schriftkultur hervor, die als solche und jenseits der ökonomischen Verwertbarkeit ihrer Texte bislang kaum Beachtung gefunden hat. Dabei scheint evident, dass in nunmehr 180 Jahren⁴ Programmiersprachen- und 80 Jahren Computergeschichte etliche dedizierte Schriftkulturen entstanden sein müssen, die ihre eigenen Sprachen, Formen, Stile und Publikationsarten und -kanäle hervorgebracht haben.

Mein folgender Beitrag will dieser Evidenz nachgehen und Programmiersprachen und Programme als Elaborate einer Textkultur verstehen, die nicht nur mit informatischen, sondern auch mit philologischen Methoden (er)fassbar gemacht werden können und sollten. Am Beispiel der in den 1970er und 1980er Jahren am weitesten verbreiteten

-
- 1 David J. Bolter: Das Internet in der Geschichte des Schreibens. In: Stefan Münker und Alexander Roesler (Hg.): *Mythos Internet*. Frankfurt a.M.: Suhrkamp 1997, 37–55, hier: 37.
 - 2 Zur irreduziblen anderen Hälfte vgl.: Friedrich Kittler: Hardware. Das unbekannte Wesen. In: Sibylle Krämer (Hg.): *Medien, Computer, Realität: Wirklichkeitsvorstellungen und neue Medien*. Frankfurt a.M.: Suhrkamp 1998, 119–132.
 - 3 Zum medienästhetischen und -technischen Begriffspaar Unterfläche/Oberfläche vgl. Frieder Nake: Das doppelte Bild. In: Margarete Pratschke (Hg.): *Bildwelten des Wissens. Kunsthistorisches Jahrbuch für Bildkritik*. Band 3, Nummer 3: Digitale Form. Berlin: Akademie-Verlag 2005, 40–50.
 - 4 Setzt man die Ausführungen in Ada Lovelaces »Notes«, in denen sie einen Algorithmus zur Berechnung der Bernoulli-Zahlen für eine künftig zu bauende Rechenmaschine entwirft, als deren Beginn (vgl. Tanja Rahneberg: Ada Lovelace und das erste Computerprogramm der Welt. In: *Max-Planck-Gesellschaft*, htps://bit.ly/3Xirkf3 [21.06.2023, 16:00]).

Programmiersprache BASIC soll gezeigt werden, wie sich Erkenntnisse aus der Lektüre von Programmcodes über diese Kultur und ihre Praktiken gewinnen ließen. Nach einer einführenden Betrachtung über die Attribuierung von Programmiersprachen als Sprachen findet eine philologische Annäherung an Sourcecodes auf Basis sprach- und literaturwissenschaftlicher Einordnungen statt. Die Betrachtung steuert auf das Ziel zu, Programmiersprachen, Programmierer:innen und die zugehörigen Computer als kulturelle Artefakte zu markieren, deren Schrifttümer es mit spezifischen Mitteln zu erforschen und bewahren gilt.⁵ Anhand von Archivmaterialien sollen dabei die spezifischen philologischen Aspekte von Sourcecodes vorgeführt werden.

1. Die Sprache(n) der Medien

Der häufig anzutreffenden⁶ Aussage, Programmiersprachen seien gar keine Sprachen, lässt sich bei genauerem Hinsehen widersprechen. Der Eindruck eines *kategorialen* Unterschiedes entsteht vor allem dadurch, dass 1. Programmiersprachen nicht von Menschen gesprochen und 2. natürliche Sprachen von Maschinen nicht verstanden würden. Das Missverständnis scheint in den Begriffen »sprechen« und »verstehen« zu liegen, die eine homozentristische Perspektive insinuiert, die in beiden Fällen meint, dass Kreativität in der Produktion und Rezeption von Sprache an den engen Grenzen maschineller Sprachverarbeitung scheitern müsse. Allerdings findet derzeit nicht nur eine ›Quantifizierung‹ dieses Verstehensbegriffs statt, bei dem die Emergenz von Bedeutungen als Statistik in neuronalen Netzen emuliert werden kann; Menschen haben auch immer schon mit nicht-natürlichen Sprachen kommuniziert. Nicht nur Maschinen tauschen sich (über technische

5 Das dahinter stehende Vorhaben und sein Ziel werden von mir derzeit im Rahmen eines Forschungsprojektes an der Universität Bonn verfolgt. (<http://txt3.de/basic> [zuletzt abgerufen am 21.06.2023]).

6 Z. B.: Gero von Randow: Reden Sie mit Ihrem Computer? In: *Zeit Online*, 04.06.2014, <https://bit.ly/44pRIGj> (03.07.2023, 9:45).

Kanäle und standardisierte Datenprotokolle) seit Anbeginn des Computerzeitalters aus; auch Menschen kommunizieren mit Maschinen formal (programmieren) wie informell (Chatbots) und verständigen sich sogar untereinander mit nicht-natürlichen Sprachen⁷ – anders wäre die heute übliche gemeinschaftliche Entwicklung von Software (aber auch von mathematischen Theorien und Beweisen) kaum möglich. Dass es sich bei Programmiersprachen zumeist um geschriebene (oder gezeichnete⁸) Sprachen handelt, ist ebenfalls kein hinreichendes Argument gegen ihre Sprachhaftigkeit, gibt es doch auch natürliche Sprachen, die nicht gesprochen, sondern nur geschrieben oder gestikuliert werden.⁹ Und schließlich kann das von v. Randow geäußerte Kriterium, dass sich mit Programmiersprachen keine Gefühle ausdrücken ließen, zumindest angezweifelt werden.¹⁰

-
- 7 Im Rahmen meiner Forschung untersuche ich beispielsweise Leserbriefkommunikation in Computerzeitschriften (die heute in Internet-Foren weitergeführt wird) und finde immer wieder Beispiele, in denen die Kommunikate der menschlichen Schreiber:innen/Leser:innen nur noch im Austausch von Code-Fragmenten bestehen, die von den Kommunikationspartner:innen als Propositionen aufgefasst werden.
 - 8 Schaltpläne, Flussdiagramme und diagrammatisch angelegte Programmiersprachen (wie Konrad Zuses »Plankalkül«, vgl. Raul Rojas: *Konrad Zuse's Early Computers. The Quest for the Computer in Germany*. Berlin: Springer 2023, 147–170 [im Druck]) müssen als (teilweise) nicht symbolische Sprachen in diesen Kontext einbezogen werden, selbst wenn sie (noch) nicht direkt auf Maschinen implementiert und von diesen ausgeführt werden können. (Vgl. Stefan Höltgen: *Open History. Archäologie des Retrocomputings*. Berlin: Kadmos 2022, 155f.)
 - 9 Hier können als Beispiele für ausschließlich geschriebene Sprachen die so genannten »toten Sprachen«, die nur noch in schriftlichen Dokumenten belegt sind, dienen – aber auch Systeme, wie die Gebärdensprache oder grafische Sprachen wie Bliss Symbolics (<https://www.blissymbolics.org/>, [zuletzt abgerufen am 03.07.2023]) und Emoticons. Vgl. <https://de.babbel.com/de/magazine/was-sind-programmiersprachen> (zuletzt abgerufen am 03.07.2023).
 - 10 G. von Randow selbst hat in einem früheren Beitrag über das (un)mögliche Bewusstsein von Maschinen eine skeptizistische Gegenfrage gestellt: »[W]oher wissen wir so genau, daß Menschen Bewusstsein haben? Vielleicht, weil sie es bekunden?« (Gero von Randow: *Roboter. Unsere nächsten Verwandten*. Reinbek: Rowohlt 1997, 247f.) Ein Programm mit Ausgabe »Ich habe Gefühle.« müsste

Die Existenz einer pragmatischen Ebene von Programmiersprachen impliziert noch weitere hierfür notwendige linguistische Kategorien, die die Bedingung für ihre Zugehörigkeit zum Phänomenkreis *Sprache* bilden: Mit ihnen müssen sich *kohärente* Formen konstruieren lassen, die sich über *syntaktische* Strukturen (Regeln) zu komplexeren, *semantisch* zusammengehörigen Gebilden kombinieren lassen. Sie müssen *produktiv*¹¹ sein, so dass sich beliebig viele solcher Sprachgebilde formulieren und variieren lassen. Mit ansteigender Komplexität der sprachlichen Elaborate (Zeichen-Wort-Satz-Text-Genre-Gattung) können *stilistisch* komplexere Varianten derselben Grundaussage und schließlich sogar *Soziolekte*¹² entstehen, die vom Wissen, dem kulturellen Kontext und der historischen Einordnung der Sprachnutzer:innen abhängen.

2. Computerphilologie

Im Rahmen eines Forschungsprojektes zur Archäologie der frühen Mikrocomputer und ihrer Programmierung¹³ habe ich u.a. einen Ansatz zur sprach- und literaturwissenschaftlichen Analyse von Programmiersprachen entwickelt. Dazu habe ich Argumente für die Vergleichbarkeit und Abhängigkeit von unterschiedlichen Programmen, die ähnliche

demzufolge zumindest Skepsis über die mögliche Gefühlsfähigkeit des Computers, auf dem es läuft, auslösen.

- 11 Das Kriterium der Produktivität meint zunächst die Herausbildung neuer Begriffe auf der morphologischen Ebene. Auch dies wird häufig als Argument gegen die Sprachhaftigkeit von Programmiersprachen herangezogen (vgl. Umberto Eco: *In Search for the Perfect Language*. Oxford/Cambridge: Blackwell 1995, 311), obgleich konkatenerative Programmiersprachen wie Forth oder Logo gerade darin bestehen, neue »Wörter« (<https://www.forth.com/starting-forth/11-forth-compiler-defining-words/>, [zuletzt abgerufen am 03.07.2023]) mit komplexeren Bedeutungen herauszubilden.
- 12 Vgl. F. Naz/). E. Rice: Sociolinguistics and programming. In: *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. Victoria 2015, 74–79. Hier werden mithilfe von KI-Verfahren stilistische Elemente in Sourcecodes gesucht, die sich als Gendermarkierungen verstehen lassen.
- 13 Vgl. Höltgen: *Open History*, 129–169.

audiovisuelle Ausgaben erzeugt haben, gesammelt. Die bloße Ähnlichkeit der Oberflächen korrespondiert nicht notwendig auch mit einer Ähnlichkeit ihrer Unterflächen. Im Gegenteil: Auf der Code-Ebene zeigen sich bisweilen kategorische Differenzen in der Art und Weise der verwendeten Programmiersprachen (BASIC, C, verschiedene Assemblersprachen) und Programmiersysteme (Digitalcomputerprogramme, Analogrechnerprogramme), die sich allerdings dennoch auf Verwandtschaften (im Sinne einer *Rezeptionsästhetik*) untersuchen ließen. Hierzu musste ich jedoch zunächst die grundlegenden linguistischen Eigenschaften, die einen solchen Vergleich ermöglichen, herausarbeiten. Dies geschah durch die Analyse von Kohäsionsverfahren¹⁴ und in der Suche nach Merkmalen produktiver Rezeption (hier: von verwendeten Algorithmen)¹⁵.

Aus den Funden und Ergebnissen ergaben sich zahlreiche Anschlussfragen, die ich als Forschungsdesiderat derzeit in einem Projekt über die Programmiersprache BASIC und ihre Programmierkulturen untersuche. Bevor ich diese Fragen und die aus den Philologien eingesetzte Methodologie vorstelle, sollen die Eigenschaften der Programmiersprache BASIC und ihrer Programmierung, die diese Sprache besonders attraktiv für eine solche Untersuchung machen, kurz skizziert werden.

2.1 BASIC

BASIC (Beginner's All-purpose Symbolic Instruction Code) entstand 1964 in den USA als Programmiersprache für Studierende der Geisteswissenschaften. In BASIC wird weitgehend von mathematischen und informatischen Programmiersprachen-Konzepten und -strukturen abstrahiert, so dass sich die Lernenden auf die Entwicklung von Algorithmen und deren Übertragung in Programme konzentrieren konnten. Die Entwickler von BASIC stellten ihre Sprache der Allgemeinheit zur

14 Höltgen: *Open History*, 147–155.

15 Höltgen: *Open History*, 132–138.

Verfügung, was nicht nur dazu führte, dass binnen weniger Jahre zahlreiche Schulen und Universitäten BASIC-Kurse anboten,¹⁶ sondern auch dazu, dass Computerfirmen ab Ende der 1960er Jahre damit begannen, eigene, speziell an ihre Systeme angepasste BASIC-Dialekte zu entwickeln.

Die Dialektvielfalt »explodierte« förmlich ab Mitte der 1970er Jahre, als Mikrocomputer für Privatleute angeboten wurden. Jeder Hersteller verbaute in sein System einen speziell an dessen Hardware angepassten BASIC-Dialekt. Zusätzlich entstanden alternative Dialekte und Sprach-erweiterungen, mit denen man die jeweiligen Systeme nachträglich ausstatten konnte, um sie für spezifische Anwendungen zu rüsten.¹⁷ Auf diese Weise entstanden bis Anfang der 1990er Jahre¹⁸ Hunderte, wenn nicht Tausende unterschiedliche BASIC-Dialekte, für die sich jeweils eigene User-Gemeinschaften herausbildeten. Programmierer:innen dieser Dialekte tauschten Kenntnisse und Programmcodes untereinander aus, für sie und ihre Plattformen wurde Sekundärliteratur (Zeitschriften und Bücher mit Programmcodes und anderen Darstellungen) publi-

16 Die führte zur oben genannten Weitverbreitung der Sprache: »BASIC was once the lingua franca for commurucatmg with computers.« (Marc Jones Lorenzo: *GOSUB without RETURN. Between the Lines of the BASIC Programing Language*. Philadelphia/Pittsburg: SE Books 2022, 235).

17 Die Vielfalt zeigt die (unvollständige) Liste an BASIC-Dialekten in der Wikipedia (https://de.wikipedia.org/wiki/Liste_von_BASIC-Dialekten [zuletzt abgerufen am 29.06.2023]). Unter den Überschriften zu den Plattformen finden sich (einige) für diese Plattform verfügbaren BASIC-Dialekte.

18 Mit dem Erscheinen der letzten Homecomputer (die sich durch speziell für den Anschluss an heimische Medientechnologie angepasste Schnittstellen und eine Hardware- und Software-Vollausstattung auszeichneten) Anfang der 1990er Jahre endete zwar nicht die Verwendung der Programmiersprache; mit dem Verlust an Plattformen mit fest eingebauten BASIC-Dialekten diversifizierte sich der Gebrauch der Sprache jedoch so stark, dass Eingrenzungen, wie die genannten, kaum noch möglich waren. Im selben Zeitraum wurden auch Prinzipien der strukturierten Programmierung in neue BASIC-Dialekte eingeführt, die den idiosynkratischen Gebrauch der Sprache stark (zugunsten einer »informatischeren« Programmierlehre und -praxis) einschränkten, was einen weiteren Verlust an Vielfalt zur Folge hatte.

ziert, es wurden Clubs gegründet und Newsletter/Fanzines herausgegeben und Hardware- und Software-Erweiterungen für das jeweilige System entwickelt.

2.2 Quellenlage

Aus dieser Vielfalt an BASIC-Dialekten ist ein Archiv erwachsen, das als das größte und vielfältigste innerhalb der Computergeschichte gelten dürfte.¹⁹ Die im Zeitraum von 1975 bis 1995 noch weitgehend fehlende oder erst rudimentär verfügbare Vernetzbarkeit²⁰ von Computern führte dazu, dass Source Codes vorrangig auf Papier publiziert und proliferiert wurden. Neben den bereits erwähnten Zeitschriften und Büchern entstand ein großer Fundus an *grauer Literatur*, in dem ebenfalls Programmcodes in kleineren und größeren Regionen unter den Computer-Nutzer:innen kursierten. Darüber hinaus existiert (in privaten Archiven und Nachlässen) ein prinzipiell nicht zu überschauender und systematisch schwer erfassbarer Fundus an *Ephemera* (Programmausdrucke, handschriftliche Notizen) und *marginalisierten* Schriftstücken, in denen sich individuelle Zugänge zur Programmiersprache und ihrer Programmierung in besonderem Maße zeigen. Hinzu kommen Datenträger (vor allem Disketten und Kassetten), die Programmcodes enthalten.

Die aus dem Ausmaß und der teilweise prekären Quellenlage dieses Archivs erwachsenden Schwierigkeiten für die Bewahrung und Erforschung dieser Textbestände werden noch durch die Tatsache verschärft,

19 Michel J. Halvorson beschreibt diese Amateur-Programmierkultur im Kapitel »Four Million BASIC Programmers«, in: Ders.: *Code Nation: Personal Computing and the Learn to Program Movement in America*. Kentfield: ACM Books/Morgan & Claypool 2020, 127–165.

20 Vernetzbar waren Computer in diesem Zeitraum entweder physikalisch durch den Aufbau von Kabelnetzwerken (was für private Zwecke eher unüblich war) oder via Telefonnetz mit Modems oder Akustikkopplern, was für Privatanwender einerseits kostenintensiv war und andererseits Einschränkungen im Datenumfang und der Übertragungsgeschwindigkeit mit sich brachte, so dass ein Austausch von Wissen, Programmen und Daten, wie er heute üblich ist, auf diesem Weg kaum stattfand.

dass die Quellen in unterschiedlichsten Erhaltungsgraden vorliegen und die auf Papier gespeicherten Informationen zudem unterschiedliche Druckqualitäten aufweisen. Und als genüge dies noch nicht, so bekommt es die Forschung zudem mit Zeichensätzen und Typografien zu tun, die spezifisch für die jeweilige Computerplattform sind²¹ und eine Digitalisierung mit Texterkennung (OCR) verkomplizieren. Hier kann zunächst allein ein menschliches Close-Reading der Source Codes erfolgen, das – ausgestattet mit Hintergrundwissen über die jeweilige Plattform und den verwendeten BASIC-Dialekt – Zugänge zur Funktionsweise, zum Aufbau und dann zum Programmierstil aus den Sourcecodes kondensiert. Welche philologischen Kategorien hierbei Berücksichtigung finden, um die anvisierten Ziele des Projektes zu erreichen, soll nun dargelegt und mit Beispielen aus meinem Archiv illustriert werden.²²

-
- 21 Anfang der 1960er Jahre einigte sich ein internationales Konsortium auf die Standardisierung von Computerzeichensätzen, womit die Vernetzung unterschiedlicher Plattformen vereinfacht werden sollte. Das ASCII-System (American Standard Code for Information Interchange) beschreibt die ersten 128 Zeichen (7 Bit), die einige Steuerzeichen, das groß- und kleingeschriebene Alphabet, die Ziffern und Satzzeichen umfassen. Homecomputer nutzen diese ASCII-Zeichenkodierung – aber zusätzlich auch noch das 8. Bit, das es ermöglicht weitere 128 Zeichen zu diesem Standardzeichensatz hinzuzufügen. In diesen werden systemspezifische Zeichen, grafische Elemente, internationale Symbole und Steuercodes hinterlegt, die von Plattform zu Plattform unterschiedlich ausfallen und oft nur (wenn überhaupt) von systemeigenen Druckern ausgedruckt werden können.
- 22 Da die Fragestellung einerseits synchrone Aspekte der Programmiersprache/Programme umfasst (Paratextualität, Individualstil, Textgestaltung, Soziolekt etc.), andererseits genealogische (diachrone) Perspektiven auf Programmiersprache/Programme beleuchtet (Dialektologie, Rezeption, Stematologie, Intertextualität etc.) und schließlich transzendente Elemente des Sprachdesigns fokussiert, werden innerhalb meines Forschungsprojektes bei jeder Kategorie alle drei Perspektiven berücksichtigt. Hier kann dies nur in Ansätzen geschehen.

2.3 Sprachwissenschaftliche Perspektiven

Von den in der Sprachwissenschaft untersuchten Kategorien *Laut*, *Wort*, *Satz*, *Text* sowie *semantische* und *kommunikative* Aspekte entfallen bei der Untersuchung von Sourcecodes die ersten beiden, denn Programmiersprachen werden nicht gesprochen, und bei der Entwicklung eines Programms spielen die morphologischen Eigenschaften der Programmierbefehle keine Rolle.²³ In Hinblick auf *klassische Grammatik* lässt sich zunächst zeigen, dass in BASIC alle Befehle als englische Verben im Imperativ vorliegen.²⁴ Interessanter zeigt sich die Konstruktion von Programmzeilen,²⁵ weil hierbei bereits *syntaktische Strukturen* zum Tragen kommen, die später dafür sorgen, dass die Programmzeile vom BASIC-Interpreter übersetzt werden kann.²⁶ Aber nicht nur die Lesbarkeit des Codes durch die Maschine wird durch diese syntaktischen

-
- 23 Dies ändert sich beim Sprach-Design, wo neben der syntaktischen die morphologische Ebene grundlegend ist.
- 24 Dies gilt auch für Variablenzuweisungen wie $X=X+1$, denn hier wäre eigentlich ein (oft fakultativ zu schreibendes) LET davor zu setzen, welches der Zuweisung überhaupt erst ihren Sinn verleiht, indem es den Gleichheitsoperator (=) zu einem Zuweisungsoperator (:=) macht. (Vgl. John G. Kemeny/Thomas E. Kurtz: *Back to BASIC. The History, Corruption, and Future of the Language*. Reading u.a.: Addison-Wesley 1985, 59f.)
- 25 Diese setze ich hier als Analogon zu den *Sätzen* natürlicher Sprache, betone aber, dass eine Übertragbarkeit der genannten Kategorien von natürlichsprachlichen auf Programmierertexte weder 1:1 möglich ist, noch dem Wesen von Source Codes gerecht würde.
- 26 Die BASIC-Dialekte der Homecomputer liegen als Interpreter vor: Hierbei wird der Sourcecode erst während der Ausführung (zur Laufzeit) in Maschinencode übersetzt, welchen der Computer dann ausführen kann. Bei Compilern wird zunächst der komplette Sourcecode in Maschinensprache übersetzt und als ausführbares Programm abgespeichert. Diese Verfahren der (maschinellen) Übersetzung bergen weitere interessante philologische Aspekte, die teilweise von der Informatik erforscht werden. Das Fachgebiet der Theoretischen Informatik untersucht in seiner Automaten-Theorie die Frage, wie Berechenbarkeit und formalsprachliche Beschreibung (also Grundlage von konkreten Programmiersprachen) in Zusammenhang stehen. Vgl. Robert Floyd/Richard Beigel: *Die Sprache der Maschinen*. Bonn u.a.: Thomson 1996.

Vorgaben ermöglicht, sondern auch die derjenigen Menschen, die diesen Code schreiben und lesen. Empirische Studien des *Code Reading*²⁷ haben hier vor allem die Rolle der Kohäsionsverfahren hervorgehoben, die syntaktische Strukturen und semantische Zusammenhänge zwischen einzelnen Programm-Teilen erzeugen.

Kohäsion (und die daraus resultierende *Kohärenz*) sind bereits textlinguistische Phänomene, die Beziehungsstrukturen über den gesamten Programmtext hinweg beschreiben.²⁸ Ihre Untersuchung ermöglicht den Nachvollzug des Programmaufbaus und liefert Anhaltspunkte über seine ›Klarheit‹ im Sinne des Programmierstils.²⁹ Die Kohärenz eines Sourcecodes konstituiert das symbolische Gebilde schließlich als einen zusammenhängenden Text, der sich von anderen Texten unterscheiden lässt und sich als Ganzes *literaturwissenschaftlich* in größere Kategorien (Genres, Gattungen etc.) einordnen lässt.

Intertextualität (in ihrem sprachwissenschaftlichen Verständnis) »bezeichnet als theoretischer Begriff [...] das Phänomen einer wie auch immer festzulegenden Relation zwischen [unterschiedlichen, S. H.] Texten.«³⁰ Diese Relationen sind unterscheidbar in Hinblick auf ihren *Grad* (von der Allusion bis hin zum wörtlichen Zitat) und der räumlichen und zeitlichen *Distanz* zum referenzierten Fremdtext. Die linguistische Intertextualitätsforschung liefert für eine Analyse Maße für die Vagheit/Konkretheit und die Enge/Weite der Referenzbeziehungen. Bei BASIC-Sourcecodes lassen sich vielfältige Formen von Intertextualitätsbeziehungen finden und kategorisieren. Dies beginnt bereits damit, dass ein

27 Vgl. Teresa Busjahn/et al.: Eye movement in code reading: relaxing the linear order. In: *Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension*, 255–265, sowie Höltgen: *Open History*, 170–172.

28 Vgl. Robert-Alain de Beaugrande/Wolfgang Ulrich Dressler: *Einführung in die Textlinguistik*. Tübingen: Niemeyer 1981, 50–117.

29 Vgl. Paul Nagin/Henry F. Ledgard: *BASIC with Style. Programming Proverbs. Principles of Good Programming with Numerous Examples to Improve Programming Style and Provicency*. Rochelle Park: Hayden.

30 Susanne Holthius: *Intertextualität. Aspekte einer rezeptionsorientierten Konzeption*. Tübingen: Niemeier 1993, 29.

BASIC-Befehl die hochsprachliche Referenz auf ein Maschinenspracheprogramm ist, welches ausgeführt wird, wenn der BASIC-Befehl eingegeben worden ist. Das Einbinden³¹ von weiteren Programmteilen/Daten in das eigene Programm (etwa durch das Nachladen eines Datenträgers) stellt eine weitere räumlich enge Referenz dar. Für eine Untersuchung der BASIC-Programmierkulturen sind jedoch vor allem räumlich und zeitlich weiter reichende Referenzen interessant. Diese zeigen sich etwa in der *Übernahme* oder *Adaption* fremder Code-Bestandteile in das eigene Programm. Die Existenz von Standardalgorithmen³² insinuiert bereits, dass es sich hierbei nicht notwendigerweise um eine plagiatorische Praxis handelt: Bestimmte, häufig genutzte Algorithmen (etwa zum Sortieren von Daten oder zum Finden von Informationen in einem Datenbestand) sind Gegenstand von Lehrwerken³³ und dürfen in eigenen Programmen verwendet werden.

Schließlich können für Programmiersprachen auch Aspekte von *Pragmatik* untersucht werden. Hier sind weniger die Sprachhandlungen (die Programme gegenüber Computern in jeden Fall darstellen: als direktive Akte) von Interesse als die Untersuchung von Implikaturen, Präsuppositionen und Methoden der Gesprächsanalyse, mit denen geklärt werden könnte, welche Rolle *Programmiersprachen in der zwischenmenschlichen Kommunikation* spielen und wie sie kommuniziert werden. Hierfür existieren interessante Beispiele, die zeigen, wie BASIC-Programmierer:innen über und mit Code kommunizieren – etwa in Zeitschriften in den Leserbrief-Rubriken. Über solche Code-Konversationen baut sich ein kulturelles Netzwerk auf, dessen sprachliches Medium BASIC und seine Dialekte sind.

31 Ein markanter Begriff der Buchherstellung, der hier als *terminus technicus* in die Praktische Informatik eingewandert ist.

32 Ein Begriff aus der Mathematikdidaktik, der standardisierte Rechenwege beschreibt. In der Informatik gibt es Bibliotheken mit Standardalgorithmen für ganz unterschiedliche Probleme, die in unterschiedlichen Sprachen z.B. hier gesammelt werden: <https://bit.ly/3JFmcMB> (zuletzt abgerufen am 03.07.2023).

33 Für BASIC ist hier etwa das zweibändige Werk von F. R. Ruckdeschel: *BASIC Scientific Routines*, Vol. 1 & 2. Peterborough: Byte/McGraw-Hill 1981 einschlägig.

2.4 Literaturwissenschaftliche Perspektiven

Intertextualität wird auch innerhalb der Literaturwissenschaften untersucht – hier allerdings weniger formal als in der Textlinguistik. Vielmehr hat die poststrukturalistische und postmoderne Literaturtheorie das Vorliegen von Intertextualität als Argument für die grundsätzliche Vernetztheit von Sprache³⁴ herangezogen. Den diesbezüglich eher sprachphilosophisch gehaltenen Ausführungen der Diskursbegründer:in Julia Kristeva³⁵ und Jacques Derrida³⁶ stellt ab Mitte der 1980er Jahre der französische Literaturhistoriker Gerard Genette eine detaillierte und strukturierte Phänomenologie literarischer Text-Text-Beziehungen, die er als *Paratextualität*³⁷ bezeichnet und von welcher Intertextualität³⁸ einen Sonderfall bildet, gegenüber. Die Beziehungen, die ein Sourcecode zu anderen Texten haben kann, sind auch hier vielfältig. So finden sich *Epitexte* (die einleitenden Sätze, die Überschriften, Kommentare der in Zeitschriften abgedruckte Sourcecodes usw., vgl. Abb. 1) und *Peritexte* (nachfolgende Leserbriefe, Errata, spätere Code-Versionen usw.). Auch solche paratextuellen Beziehungen von Sourcecodes liefern Argumente für eine kulturelle Auseinandersetzung mit Programmiersprachen und Computerprogrammen.

-
- 34 Petra Steiner: *Intertextualität*. In: Stefan Höltgen/Patrick Baum (Hg.): *Lexikon der Postmoderne. Von Abjekt bis Zizek*. Bochum: Projektverlag 2010, 99–101.
- 35 Julia Kristeva: Wort, Dialog und Roman bei Bachtin (1967). In: Jens Ihwe (Hg.): *Literaturwissenschaft und Linguistik. Ergebnisse und Perspektiven*. Band 3: Zur linguistischen Basis der Literaturwissenschaft II. Frankfurt a.M.: Athenäum 1972, 345–375.
- 36 Jacques Derrida: Die Differance. In: Ders.: *Randgänge der Philosophie*. Hg. v. Peter Engelmann. Passagen: Wien 1988, 29–52.
- 37 Gerard Genette: *Paratexte. Das Buch vom Beiwerk des Buches*. Frankfurt a.M.: Suhrkamp 2001.
- 38 Gerard Genette: *Palimpseste. Die Literatur auf zweiter Stufe*. Frankfurt a.M.: Suhrkamp 1993, 10f.

Abb. 1: Sourcecode mit Paratexten in einer Computerzeitschrift.³⁹



Eine Weiterung der linguistischen Betrachtung stellt die Einordnung des Textes als *Textsorte* dar und kaum etwas spricht dagegen, Sourcecodes als dezidierte Textsorte zu behandeln: Sowohl textinterne Kriterien (Lexik, Syntax, Textaufbau und grafische Anmutung) als auch textexterne Kriterien (Funktion, Umfang, Medium, das den Text enthält, situative Einbettung) sind weitgehend spezifisch für Programmcodes.⁴⁰ Daher wäre in der Folge zu fragen, ob sich neben der texttypologischen auch weiterreichende *literaturwissenschaftliche Klassifikationen* zur Einordnung anbieten – etwa ob es *Gattungs-* oder *Genre-*Einteilungen geben

39 Jochen Hartwig: Hinterhalt. In: HC – Mein Home-Computer (12/1983), 43. Hier zeigen sich auch die typische Druckqualität und die Verwendung nicht-standardisierter Sonderzeichen.

40 Daniel Siebrecht: Textsorten im Informatikunterricht. In: https://ddi.uni-wuppertal.de/website/repoLinks/v50_vortrag.pdf (zuletzt abgerufen am 29.06.2023), 15f. Die Typologie von Textsorten, die zumeist nach ihrer Funktion geordnet werden, bleibt für Programmtexte allerdings ebenso unscharf wie für natürliche Sprachen (vgl. de Beaugrande/Dressler 1981: *Textlinguistik*, 189–193), denn die Ausführung eines Programmtextes durch einen Computer ist zwar seine »dominante Funktion« (ebd., 191), aber nicht die einzige.

könnte, die im Zusammenhang oder im Kontrast zu Oberflächen-Beschreibung (»Computerspiel«, »Adventure«) stehende Kategorien anbieten. Hier wären für BASIC-Programme etwa der Zeilenumfang des Codes⁴¹ (vgl. Abb. 2), die Einbettung von maschinennahen Operationen im Code⁴² oder die Strukturiertheit des Programms⁴³ Kriterien für eine Zuordnung.

Abb. 2: Ein »BASIC-Oneliner«-Programmcode.⁴⁴

```
1 BORDER 0:POKE 23693,1:CLS :DIM P(10):FOR L=0 TO 4:RESTORE :REA
D W,S,M$:FOR T=1 TO 10:LET P(T)=85-9*T:PRINT AT T,0;M$(T*W-11
TO T*M):NEXT T:LET M$(P(1)+1)="":LET K=113-CODE INKEY$:FOR A=1
TO L+2:LET G=P(A):LET I=4*RND-2:LET I=(A=1)*((K=1)-(K=2))+M*((K=1
G)-(K=0))+SGN I*(1+1*(I*I<1))*G<>P(1)):LET C=CODE M$(G+I+1)-4
5:LET P(A)=G+I*(C<2)-I*(M$(G-1+1)<"0")*(C>1)*(A>1):PRINT AT G/W+
.5,G-M*INT (G/W);M$(G+1);AT W,0;L*50+S;AT P(A)/W+.5,P(A)-M*INT (
P(A)/W);INK 7-A*(T<1);"00000"(A):LET T=T-1+(C=-9*A)*(37-T):LET
S=S+(A=C):IF A=1 OR P(A)-P(1) OR T>0 THEN NEXT A:POKE (S<56)*23
620,M:NEXT L: DATA 12,1,"000000000000...0.....00.0..00.00.00
0...$.0.00....00....",M$+M$(W TO )+M$
```

-
- 41 Hier existieren zum Beispiel etwa »Zwanzigzeiler« oder »One-Liner« als Programmierherausforderungen.
- 42 Bestimmte BASIC-Befehle erlauben Hardware-Zugriffe. Diese reichen von Manipulationen einzelner Speicheradressen bis hin zur Implementierung von Maschinensprache-Programme mithilfe von BASIC-Laderoutinen.
- 43 Die mögliche Unstrukturiertheit von BASIC wurde von der Informatikdidaktik früh kritisiert (vgl. Arbeitskreis Schulsprache: Empfehlung, Ergebnisse, Abschlussbericht und Stellungnahme zu BASIC. Paderborn: FeoLL 1976, C1-C14). Spätere Lehrbücher zur strukturierten BASIC-Programmierung hatten das Ziel, leichter les- und wartbaren Sourcecode zu ermöglichen (zum Beispiel: Johann Weilharter: Spaß mit Algorithmen. Einführung in das strukturierte Programmieren mit 42 BASIC-Programmen. Wiesbaden: Springer 1984).
- 44 Dr. BEEP: *Tiny Pacman*. In: *RETRO – BASIC-Sonderheft* (2013), 15; ebenso in: <https://rtrto.de/tinypacman> (zuletzt abgerufen am 03.07.2023). Dieses Programm implementiert mit einer einzigen BASIC-Programmzeile eine Variante des Spiels »Pac-Man« auf dem ZX-Spectrum-Computer.

Zwei zentrale Punkte der Erforschung von Programmierkulturen liegen in der *Editionsphilologie* einzelner Programmcodes und in der *Rezeptionsgeschichte/Stemmatologie* spezifischer Code-Elemente und -Genres. Die editionsphilologische Perspektive nimmt die Genealogie eines einzelnen Programmcodes in den Blick und versucht über das *Versioning* aus den Codefragmenten Aspekte informatischer Selbstausbildung und algorithmischen Denkens der Programmierer:innen herauszulesen. Ein Computerprogramm, zumal wenn sein Sourcecode umfangreich ist, wird nicht linear und selten an einem Tag entwickelt. Zur Sicherung werden die Zwischenstände (manchmal unter Angabe der Versionsnummer oder des Datums im Dateinamen) abgespeichert, um sie später wieder laden und weiterentwickeln zu können. Vereinzelt existieren von Programmen noch die Entwicklungsversionen – gespeichert auf Datenträgern oder als Ausdrucke, die im Idealfall handschriftliche Korrekturen und Annotationen des/der Programmierer:in enthalten. Der kritische Vergleich solcher Versionen mit dem finalen Ergebnis kann wertvolle Informationen über die Entstehungsgeschichte des Programmtextes liefern. Dort, wo diese Sicherungen nicht existieren, geben die Sourcecodes manchmal Hinweise auf ihre Bearbeitungsstufen. (Vgl. Abb. 1, wo die BASIC-Zeilen 15 und 405 offenbar nachträglich eingefügt wurden, während die Zeilen 60, 80, 260 und ggf. weitere nachträglich gelöscht worden sein könnten.)

Wie bereits beim Thema Intertextualität angesprochen, finden nicht selten Übernahmen von fremden Codefragmenten (oder auch noch nicht kodierten Standardalgorithmen) in das eigene Programm statt. Die *historische Rezeptionsästhetik* – und dort vor allem die Ausführungen zur »produktiven Rezeption«⁴⁵ – liefern Ansätze, die es ermöglichen, Autor:innen-Leser:innen-Beziehungen zu rekonstruieren. Aus konkreten Übernahmen und mutmaßlichen Ähnlichkeiten

45 Gunter E. Grimm: *Rezeptionsgeschichte. Grundlegung einer Theorie*. München: Fink 1977, 147–153. Wenngleich auch hier der Vorwurf von Vagheit bei der Zuschreibung im Raum steht. (Vgl. Maria Moog-Grünwald: Einfluss- und Rezeptionsforschung. In: Manfred Schmeling (Hg.): *Vergleichende Literaturwissenschaft. Theorie und Praxis*. Wiesbaden: Athenaion 1981, 49–72, besonders: 58–65.)

lassen sich diachrone Vernetzungen in Programmierkulturen ablesen und aus Abweichungen und Änderungen übernommener Vorlagen die epistemologische und ästhetische Progression spezifischer Programme und ihrer Codes ablesen. Wie oft kritisiert⁴⁶, sind die Analysekriterien hierfür nur vage formuliert und belegbare produktive Rezeption eher die Ausnahme. Dennoch kann der Blick aus dieser Perspektive interessante Hinweise für die Eingrenzung von Coding Communities (in denen dieselben »Urtexte« kursieren) erbringen. Das aus der diachronen Editionsphilologie stammende Programm der *Stemmatologie* sucht dabei nach Anhaltspunkten für die wahrscheinlichsten Überlieferungsabhängigkeiten. Hier kommen bereits computerisierte Verfahren der Identifikation von Übernahmen zur Anwendung,⁴⁷ die sich (etwa im Rahmen eines Digital-Humanities-Forschungsprogramms) auch für Sourcecodes anbieten.

2.5 Bibliothekswissenschaftliche Perspektiven

Schließlich müssen hier noch einige Überlegungen aufgeführt werden, die sich mit der Materialität und Nutzung von Quellen befassen. Da ein Großteil der BASIC-Sourcecodes auf Papier vorliegt, erscheint eine Kategorisierung dieser Quellen sinnvoll. Sie kann einerseits Hinweise auf die Diskursivität von Sourcecodes geben (diese reicht von internationaler Verbreitung bis hin zu alleiniger Nutzung durch den/die Autor:in). Andererseits hilft eine solche Einordnung auch bei der epistemologischen Analyse der Texte oder Textfragmente.

46 z.B. Maria Moog-Grünwald: Einfluss- und Rezeptionsforschung. In: Manfred Schmeling (Hg.): *Vergleichende Literaturwissenschaft. Theorie und Praxis*. Wiesbaden: Akad. Verl. Athenaion 1981, 49–72, hier: 54f.

47 Armin Hoenen/Gerrit Brüning: Überlegungen zur Stemmatologie neuerer Überlieferungen. In: <https://t.ly/gOch> (zuletzt abgerufen am 29.06.2023).

Abb. 3: Beispiel eines privat produzierten Club-Magazins.⁴⁸

PRICE \$2.00

THE 80 NOTEBOOK

JULY 1980 ISSUE #4

.....

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

.....

NLOS/1 AND NLOS/2

A Natural Language Operating System for the TRS-80

What is natural language? For our purposes, natural language is common, ordinary English - the expression of facts using simple sentences. NLOS/1 is a system which allows the computer to "understand" the information conveyed to it through simple sentences and to answer questions concerning the information conveyed. This ability makes the system an excellent tool for the creation, management and inquiry to a conversational data base of facts and figures. It can also be an educational tool - as a study in artificial intelligence through an examination of the internal workings of the program, or by its reaction to math reading problems of various complexity. It can be an excellent tool in teaching English grammar, sentence structure and logical deductive reasoning to students, young and old alike. In any case, holding a conversation with your computer can be a lot of fun.

Let's see how the system accomplishes this and what its limitations are:

First, we have to get down to the basics of English. The system recognizes phrases grouped together in a sentence. A phrase is a group of one or more words that, together, convey a concept or identify an entity. These phrases can convey a subject, a verb, a preposition, a conjunction, a modifier or a question invoker. These represent the grammatical types that NLOS/1 can handle. You may ask - why not recognize individual words rather than phrases so that the conventional grammatical types - nouns, adjectives, verbs, adverbs, prepositions, conjunctions, interjections, and pronouns may be used? Well, recognizing nothing smaller than a phrase alleviates the problem of context usage. For example - "the President of the United States" contains adjectives, nouns and a preposition, yet the phrase identifies a single subject. Each grammatical type further conveys a type of information. Subjects identify a person (a who - "Tom", etc.), a place (a where - "New York City", etc.), or a thing (a what - "a banana", etc.).

Verbs convey the type of object clause they affect - in "Tom said nothing" and "Tom went home", etc., "said" acts on "what" and "went" acts on "where".

48 The 80 Notebook, Issue 04 (7/1980), S. 1, <https://bit.ly/3Cyoj0C> (zuletzt abgerufen am 03.07.2023). Hier wird ein Betriebssystem mit natürlichsprachlicher Ausgabe für den TRS-80-Computer vorgestellt, dessen BASIC-Listing zum Abtippen sich weiter hinten in der Ausgabe befindet. Die Druckqualität ist wie üblich schlecht; das Magazin wurde mit einem Matrixdrucker ausgedruckt und per Fotokopie vervielfältigt.

BASIC-Sourcecodes auf Papier wurden in Büchern (Handbücher zu Computern, Sekundärliteratur), Zeitschriften (Periodika, Club-Zeitschriften, Newslettern), als Computerausdrucke und handschriftlich zu Papier gebracht. Während Bücher und Zeitschriften, die durch ISBN und ISSN katalogisiert wurden, zumeist noch leicht auffindbar sind, lassen sich Publikationen der so genannten *Grauen Literatur*⁴⁹ weniger leicht in Archiven verorten. Hierzu zählen bereits die Handbücher der Computer, die oft von den Computerfirmen selbst verlegt und nur zusammen mit den Computern abgegeben wurden, aber auch von Amateuren publizierte Periodika (Club-Magazine, Newsletter, Vgl. Abb. 4). Die Bibliothekswissenschaft ist seit einiger Zeit bemüht, Katalogisierungssysteme für Graue Literatur zu entwickeln, damit die Bestände, die archiviert sind, auch erfasst und recherchiert werden können.

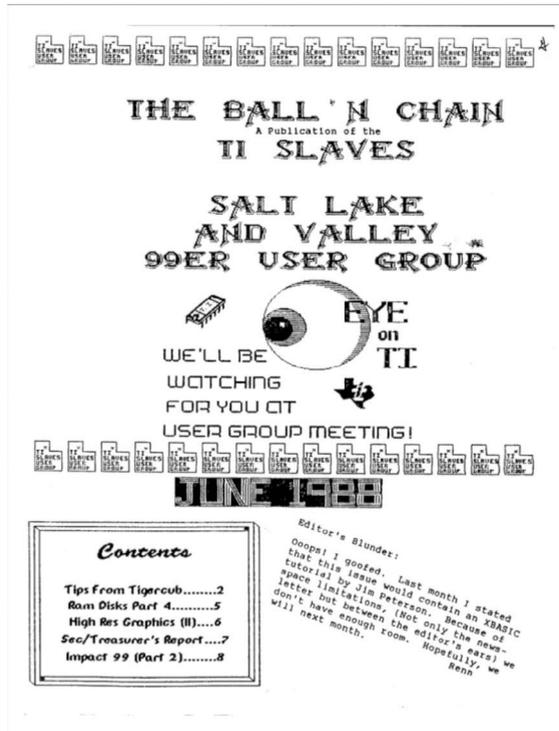
Noch problematischer sieht die Quellenlage bei so genannten *Ephemera* aus: Hierzu zählen sowohl handschriftlich notierte Sourcecodes⁵⁰ (Abb. 5) als auch Programmausdrucke (die nicht selten zum Zwecke des *Debuggings* ebenfalls mit handschriftlichen Marginalien/Annotationen versehen wurden, vgl. Abb. 6) und marginalisierte Printpublikationen. Es liegt im Wesen dieser Schriften, dass sie sich gar nicht systematisch erfassen lassen. Demzufolge ist die bibliothekswissenschaftliche Erforschung und Systematisierung solcher Quellen auch noch nicht weit

49 J. M. Gibb/E. Phillips: Bessere Zeiten für graue oder nicht herkömmliche Literatur. In: *Bibliothek* 3. 1979. Nr. 2, 122–126.

50 Hiervon gibt es im Heimcomputer-Zeitalter erstaunlich viele, was mit dem Käufer-/Nutzer-Alter der Computerbesitzer:innen zusammenhängt: Kinder und Jugendliche, die vielleicht nur sporadisch Zugang zu Computern hatten und Codes daher handschriftlich notiert haben, um sie später in den Computer eingeben zu können. Diese (Hand-)Schriften zählen zu den informatikdidaktisch interessantesten Quellen, weil sie das algorithmische Denken der Autor:innen ungefiltert vor Augen führen.

fortgeschritten⁵¹, wenngleich erste Ansätze (auch zu Computercodes⁵²) vorliegen.

Abb. 4: Deckseite des Newsletters vom TI-99-Computerclub in Salt Lake City (USA), Ausgabe 06/1988.



- 51 H. J. Jackson: *Marginalia. Readers Writing in Books*. New Haven/London: Yale Univ. Press 2001. Sowie: D. C. Greetham (Hg.): *The Margins of the Text*. Ann Arbor: The Univ. of Michigan Press 1997.
- 52 GreyNet International 1992–2017: *Document Types in Grey Literature*. <https://bit.ly/3NZfXpe> (zuletzt abgerufen am 29.06.2023).

Abb. 5: Handschriftlicher BASIC-Sourcecode mit Korrekturen⁵³



Abb. 6: Marginalisierter Programmausdruck (Auszug)⁵⁴

```

13000 DEFINE PROCEDURE LABEL
13010 AT #6,0,11 : CSIZE #6,1,0 : PRINT #6,"DE(N) DE% CR%" : CSIZE #6,0,0
13020 FOR N=1 TO OB%
13030 AT #6,N,1 : PRINT #6,OBS(N)
13040 AT #5,N,1 : PRINT #5,OBS(N)
13050 END FOR N
13060 FOR N=1 TO FE%
13070 FOR M=1 TO S
13080 AT #4,M-1,(N*3)-2 : REMARK mit hier mit funktion (N*3)-2, M-1 statt
13090 PRINT #4,FES(N,M)
13100 AT #5,0,15 : CSIZE #5,3,0 : PRINT #5,"REGEL(N)": CSIZE #5,0,0
13110 END FOR M
13120 END FOR N
13130 END DEFINE LABEL
13000 DEFINE PROCEDURE ZERO_DE
14010 DE%=0
14020 ← FOR N=1 TO OB%
14030 DE(N)=0
14040 END FOR N
14050 END DEFINE ZERO_DE
15000 DEFINE PROCEDURE UPDATE_DV
15010 FOR N=1 TO FE%
15020 FOR M=1 TO OB%
15030 DE(M) = ((DE(M)*3) - ((FV(N)*RU(N,M))*3)) / 3 ? +
15040 END FOR M
15050 END FOR N
15060 END DEFINE UPDATE_DV
16000 DEFINE PROCEDURE DECISION
16010 FOR N=1 TO OB%
16020 IF DE(N) > DE% THEN DE% = DE(N) : TS% = N problem " < u. >"
16030 END FOR N
16040 END DEFINE DECISION
    
```

53 Quelle: eigenes Archiv.

54 Quelle: eigenes Archiv.

3. Schluss: Paperware und Knowledge Preservation

Die zuletzt aufgeführten Beispiele lassen den Reichtum der Quellen und die Bedeutung ihrer Inhalte vermuten, zeigen aber auch bereits die prekäre Situation, in der sich die *Paperware Preservation* befindet: Je »grauer« und ephemerer ein Schriftgut ist, desto wahrscheinlicher ist es, dass es unwiederbringlich verloren gehen könnte. Einige der abgebildeten Dokumente stammen aus Sammlungen privater Computernutzer:innen (wie auch in Abb. 7), die als Paratexte zusammen mit ihren obsolet gewordenen Computern abgegeben wurden. Es ist nicht unwahrscheinlich, dass viele solcher Schriftstücke entsorgt wurden. Diejenigen, die der Forschung zur Verfügung stehen, müssen materialgerecht bewahrt werden. Hierzu zählt auch, eine bibliografische Systematik und Möglichkeiten für ihre Digitalisierung und Archivierung zu entwickeln.

Abb. 7: Textkonvolut aus dem Nachlass eines Computerhobbyisten⁵⁵



Welches Wissen sich über die Programmierkulturen der Vergangenheit aus den Sourcecodes ziehen lässt, ist noch kaum abzuschätzen.

55 Quelle: eigenes Archiv.

Deutlich zeigt sich jedoch jetzt schon, dass die Sourcecodes nicht bloß Anweisungsketten für Computer sind, um Oberflächeneffekte in Form von Programmausgaben zu erzeugen, sondern in ihrer Unterschiedlichkeit, ihrer Individualität und ihren Beziehungen untereinander, in den individuellen Programmierstilen ihrer Autor:innen und den Publikationsformen und -medien auch komplexe Wissensspeicher darstellen. Sie lesbar zu machen – mit menschlichen und maschinellen Verfahren – öffnet der Kultur- und Technikgeschichte ab dem 20. Jahrhundert einen alternativen Blick auf den Werdegang der Digitalisierung und Algorithmisierung der Kultur. Überdies ermöglicht die philologische Erschließung der Textsorte Sourcecode interdisziplinäre Forschung zwischen den ›zwei Kulturen‹ technisch-informierter, Sourcecode-produzierender und linguistisch-informierter, quellenkritisch-lesender Disziplinen.

“I hope you can read this”

Uncovering messages with Critical Code Studies

Mark C. Marino

“[I hope you can read this.]”

That is the first line of the source code of the work of interactive fiction called *The Gay Science* by Capricorn Van Knapp. The title of the work comes from Nietzsche's *Die fröhliche Wissenschaft*, and the piece offers a meditation on loss and the eternal return (*Ewige Wiederkehr*).¹ Let us read this code together as a way of pursuing the question: what does our interpretive exploration offer? The code goes on from there:

“[I hope this works.]

[I hope.]

‘The Gay Science’ by Someone Who Loves You.

[This is the truth, as I understand it.]”

The lines in brackets are commented out in the interactive fiction programming language Inform 7. The first line that is functional code (as opposed to comments) is the title and author credit, although note that

1 Christian Benne/Jutta Georg (Hg.): *Friedrich Nietzsche: die fröhliche Wissenschaft*. Berlin/Boston: De Gruyter 2015. [uosc.primo.exlibrisgroup.com](https://doi.org/10.1515/9783110440300), <https://doi.org/10.1515/9783110440300>.

another name (Someone Who Loves You) stands in for the game's author (Van Knapp, which is also a pseudonym for Dan Ravipinto), so it would not be right to call that line merely a functional label, like the title of a document. It is more than merely the title of the piece and its author, it is part of the narrative of an interactive fiction framed as a work by a fictional Someone for a fictional You, reminding us that when we are reading the code we are still in the narrative framework of the story. In interactive fiction written in Inform 7, since the code shapes text that displays the story, it is difficult to draw a strong division between code that produces the story and code that tells the story, and it is especially difficult in a project written for a contest for beautiful code. More on that momentarily.

As we move sequentially through the code, after a chapter heading ("Chapter 1 – Alethiology," which I will gloss as "the study of truth"), an organizational feature of Inform 7, the code continues with more comments:

"[There is truth here. I have to believe that. For I may only make statements of truth here. For only the truth can save you.]

The story headline is 'An Interactive Truth'.

[Only the truth can set you free.]"

The "story headline" is another feature of Inform 7, which will appear as a subheading after the title, so that the start of the story will read, "The Gay Science" followed by the line "An Interactive Truth by Someone Who Loves You." The code continues.

"Section 1 – Us (You and I)

You are a man.

Someone Who Loves You is a man."

These lines create characters, whose genders may inordinately focus our attention on the homoerotic meaning of a work that is deeply concerned with Nietzsche's text. Nonetheless, this game is ultimately a love story or an infinity of love stories about two men, a "You" who is lost and a "Someone Who Loves You" who is trying to reach out to them. The names of the two men are interchangeable, as this sad romance generator randomly serves up characters coupled in the positions of You and Someone Who Loves You. The You character in each iteration is lost in some way. The Someone Who Loves You, who, as it seems, is the avatar or proxy for the fictional character author of the code, wants to rescue or at least reconnect with the You. We, readers of the code, are invited to identify with the You character, as the section header indicates, for we are part of the "us."

In generated story after story, the two characters take their position in randomly chosen (or pseudo-randomly chosen) scenarios, from lovers separated by death to a police detective romantically entangled with a suspect. That fungibility of the reiterated stories and characters play out Nietzsche's eternal recurrence, for these characters can be given randomly assigned names and fictional positions, but their pairing in this just-out-of-reach love story is the same. The code itself becomes a self-declared representation of "eternal recurrence" and our reading of the code exposes that existential pattern, exploring its repetition with randomized variation. Life is a recurrence of the same story in a randomly chosen change of clothes.

However, randomness, in this code, is not always so random.

In a gesture of love and loss, Someone, the fictional author of the code, decides to rig the odds of the game. The following code determines whether the program will display the "happy ending" or the "bad ending."

```

To say where this story ends:
  if a random chance of 1 in 1 succeeds:
    now the happiness of the end is true;
    say '[the happy ending]';

```

otherwise:
say '[the bad ending]':

[And so because I love you, I cheat. And give you only happy endings.]”

As it turns out, this game will only deliver happy endings, for the “random chance of 1 in 1 succeeds” every time. The program has been written to stifle the randomness, and that act is meaningful, that choice in the code is the sign of the programmer, Someone, placing their hand on the scale all for the sake of You.

But who is this “You” of the comments? Who is being addressed? One reading is that the You is us, anyone who reads the code of this piece, which was entered into a competition for a work of beautiful code. The first event of the Second Quadrennial Ryan Veeder Exposition for Good Interactive Fiction challenged participants to write a great game with “beautiful source code text.” And yet, like a poem, the speaker or encoder seems to be a character who is other than (but possibly a stand-in for) the code’s author, and the addressee seems to be someone that character loves or loved before they were lost to him. We are reading someone else’s love letters that take the form of a work of interactive fiction. Or we are reading perhaps the letter a lover wished he had sent, wished he could send, a letter made out of code.

As I have argued elsewhere,² the code of Inform 7 is deceptively English-like, or perhaps better said, the code looks like English sentences which may obfuscate its status as code, with natural language reserved words, tokens that are arranged in a sentence-like syntax. For example, the line “now the happiness of the end is true” seems to be telling us that there is true happiness at the end of the story. But what that code is literally doing is setting the value of “the happiness of the end” to true. In the English sentence the assertion offers a judgment. In the Inform 7, the statement makes an assignment. That difference marks a central tension

2 Mark C. Marino: *Critical Code Studies*. Cambridge MA: The MIT Press 2020, 150.

of *The Gay Science*, where the work presents love stories with happy endings but only as programmed fictions, as maquettes under the control of the programmer, who presumably, like the rest of us, has very little control over real life, and hence ultimately experiences loss. This code then presents the fantasy, the love offering, of one who longs to reconnect, who longs to program for themselves a guaranteed happy ending. We would not know any of this backstory without reading the code, for the game itself offers little sense of what is at play or even how to play it. For the most part, you have to read the code to know what to do in the game, and "reading the code" comes to stand in for a kind of philosophical awakening, but also taking a peek at the private subtext and encoded messages of the code's fictional author.

Only the code indicates how to reach the game's end. In the game, the player can eventually move through successive spaces outward that represent moving sequentially through the story, from the beginning to each "consequence" and on to the end, including a section marked the *Eternal Recurrence*. However, to finally reach an end, the player must reverse course and move "inside." That is how they find their way to the ending called "Home," just how the *You* of the story could find their way to the heart of the *Someone* if they only read the code. They must move, in a sense, "inside" the work of interactive fiction.

The *Gay Science* serves as a powerful and literalized illustration of the kinds of discovery that awaits readers who explore any source code. In any code, there are messages and meaning that a (typically) unintended audience can find.

Code is a message overheard, full of remnants of the design process and sometimes indications of future directions. These signs are not merely in the comments of the code but the code itself. Though we name what intention lay behind the choices at our peril, the question "why" may lead us to a fuller understanding of how this code came to be. The code certainly offers a sense of how this system operates, and there are further signs of "when" it came to be and "who" is behind this code. As to the "what," the code does not so much offer us the underpinnings of the meaningful software object to be interpreted, but is in itself a complementary object to be perused.

1. Why code?

Computer code exists for human readers. Computers do not need code (assuming they need anything at all). They could run, as Kittler has suggested, on electrical signals (i.e., “There is No Software”).³ Code exists so that humans can see their own inputs to the computer in the form of programs and data, a distinction that becomes blurry on further inspection.

The more obvious examples, or rather the most easily accessible examples of code analysis for readers steeped in the humanities, involve the code for aesthetic objects, such as interactive fiction and other electronic literature. Code poet and critic John Cayley recently made the pronouncement,

“We are right, critical code studies is key for the hermeneutic understanding of literary work. And we must keep on telling our colleagues in literary studies that reading the code of such work is required; or that if code reading is not done, the omission should at least be acknowledged. For true close reading, this work *must* be done.”⁴

His stance is particularly pertinent, since the original Critical Code Studies manifesto⁵ was written in response to his comments about code. Nonetheless, if we take Cayley’s words to confine critical code studies to the literary sphere, we miss a major contention of these practices, the contention that all code offers material for interpretation

3 Friedrich A. Kittler: “There Is No Software.” In: *CTheory* (1995): <http://www.ctheory.net/articles.aspx?id=74> (accessed September 24, 2023).

4 John Cayley: *Bridging Electronic Literature & Critical Code Studies (a Digital Humanities Quarterly Discussion)*. <https://ucpages.uc.pt/events/overcoming-divide-s-electronic-literature-and-social-change/programa/> (accessed September 24, 2023). Electronic Literature Organization Conference, University of Coimbra, Portugal.

5 Marino, C. “Critical Code Studies.” *Electronic Book Review*, Dec. 2006, <http://www.electronicbookreview.com/thread/electropoetics/codology> (accessed September 24, 2023).

because code is a cultural text. More importantly, code is a cultural text made out of synthetic languages with a highly unusual double status as messages to humans to read and messages for machines to process.

Nonetheless, I do not want to make too much of this work of interactive fiction, which was created in the context where human readers (the contest judge and audience) are expected to read the code. I do not want to make it seem that the only pieces of code worth reading are either the code of art objects or code written as art. However, critical code studies posits that all code can and should be interpreted and offers opportunities to explore and produce meaning.

The Gay Science makes that invitation to read code literal, but I contend that it has a lesson about every piece of code. First of all, while code is for computers, code also addresses a reader, who may be the same programmer later on or maybe others. Code reveals an orientation toward the world through a problem space and decisions that have been made. Priorities. Models. Those decisions are not merely in constructing the code but also in the choice of the language and architecture, and those choices have been made within material and historical constraints. The reader of the code has the opportunity to find a message in the code, underlying logic, as well as signs of possibility for what they may never have encountered in the program, just as reading the code may not give them a full sense of what the code does. Thus, the process of reading the code and watching it operate is dialogic. Think of the practice of putting breakpoints and flags in code. I feel invited by software to read its code though I am not the one to whom it is addressed.

The digital world is made out of code. Often invisible, computer source code constructs and transforms our world, touching almost every facet of our lives. Under the metaphor of transparency, glossy and spare interfaces hide the code from us until it becomes as invisible as the wiring in our machines and the blood vessels in our body, or perhaps more accurately, the laws that govern our nations or the social norms that control our behavior. Think of the iPhone, a sealed, sleek brick with inaccessible circuitry, appearing to be only interface. Think of the Google search page, bare white except for a single text-entry box and two buttons. However, transparency, in this context, signifies its opposite

as software functions through an opaque interface designed to create the illusion of a clean window, a bare desk, or a blank page. Only in the moments of fissure or error does any sign of code emerge. And when we do encounter code, most cannot read it, despite increasing calls for programming literacy.⁶ Actually, even the programmer working on the enterprise-level software package who inherits a project from another programmer may find that code inscrutable, as it has been constructed using just-in-time solutions with after-the-fact patches, all in desperate need of refactoring and often documentation. Yet, since code touches so many of our systems and has such a definitive effect on the way we live, not reading the code means we do not know what decisions have been made for and about us and why. Coming from a country that once rebelled against “taxation without representation,” it is ironic that we now live in an age of “computation without comprehension,” especially when you add the new affordances of LLMs to allow us to produce code without a definitive knowledge of how it works. Of course, the pragmatic understanding of code (knowing how our software works) is only the beginning of the fruits of close reading code. But placing all code into a kind of black box creates a crisis of literacy (Vee). To respond to that crisis, I offer *Critical Code Studies (CCS)*, a collection of methods and interpretive practices that have been growing for nearly 20 years, developing tools for understanding code and exploring what it means for technoculture.

Critical code studies is the interpretation of the extra-functional significance of computer source code using the hermeneutics of philosophy, or, as it is known in the humanities, critical theory. “Extra” here means not in addition to but growing out of. In other words, the meaning is not outside of the functioning of the code but something that takes its computational effects as a basis but only a starting point.⁷ A CCS reading is built on the foundation of an understanding of what the code does, the systems with which it interoperates, as well as its

6 Annette Vee: *Coding Literacy: How Computer Programming Is Changing Writing*. Cambridge MA: The MIT Press 2017.

7 Marino: *Critical Code Studies*, 39.

history and evolution. Not so much a method as a developing collection of approaches, critical code studies sees code not as the ends of the analysis but the entrypoint into discussions of the culture in which the code was written and operates. Critical code studies takes code as a "text," a "cultural text," meaning that it is an artifact that carries meaning in cultures and can be read, not that it is made out of linguistic signifiers and can be read as static. The object of study is not merely the static text of the code but the effects that it has on the system and the many states it creates when a system processes it.

From the moment I first proposed critical code studies in 2007 at the MLA convention in Philadelphia, PA, CCS created controversy. To be honest, that controversy caught me off guard. Audience members asked: Why should literature majors be studying code? Why should readings of code be discussed at a literary convention? What could we possibly say about code? It's just math, just algebra, after all, right? The field caused controversy not just in literary circles but in computer programming circles as well. Perhaps writing an article about a queer programming language, *Transcoder*, was not the mildest of beginnings.⁸ The computer language theorists of the discussion board *Lambda the Ultimate* were furious that the literature majors were coming to plant their flag on the shores of code with their ridiculous language of deconstruction and left-wing politics of Queer Theory and Feminisms (see *Why We Must Read Code*).⁹ Would they now have to suffer through discussions of their code the way they had to sit through discussions of Rilke and Shakespeare? If neither the literary scholars nor the programmers could support the endeavor, who could?

8 Mark C. Marino: *Of Sex, Cylons, and Worms: A Critical Code Study of Heteronormativity*. In: *Leonardo Electronic Almanac* Vol. 17/2 (2012): http://www.leoalmanac.org/vol17-no2-of-sex-cylons-and-worms/?utm_source=rss&utm_medium=rss&utm_campaign=vol17-no2-of-sex-cylons-and-worms (accessed September 24, 2023).

9 Mark C. Marino: *Why We Must Read the Code: The Science Wars, Episode IV*. In: Matthew K. Gold/Lauren F. Klein (ed.): *Debates in the Digital Humanities*. Minneapolis/London: University of Minnesota Press 2016, <http://dhdebates.gc.cuny.edu/debates/text/64> (accessed September 24, 2023).

So there it was. The Berlin Wall between the programmers on the one side and the literary scholars was reinforced and the division between the sides reified. The Two Worlds of C. P. Snow's formulation. In response, the early code studies scholars knew we had to build some coalitions and, I suppose, tunnel through the wall.

Thus began the Critical Code Studies Working Groups, biennial online gatherings of scholars and artists from around the world, drawn from professional programming and academia, hobbyist artists, and newcomers to the world of code. The goal was to see what could be said about code. Early readings from that group found meaning in natural language parts of code, such as method names or comments. For example, when I found that the developers of the *Transborder Immigrant Tools*, a satellite-driven GPS project named a method after water witching or water divining, I discovered cultural subterfuge that *Electronic Disturbance Theater 2* had tucked into their code.¹⁰ It was hard to ignore the reverberations of function names such as “witchingEvent,” marking when the system found nearby water. Suddenly a modern digital system seemed to be framing itself within the context of a folk practice. How could that not shape the meaning of the code and the software itself?

However, CCS theorist Evan Buswell has warned that if we only ever interpreted names of methods and variables, we would easily be dismissed as interested only in the ornamentation of code, not the tokens and methods themselves.¹¹ That is why an invitation by code poet and Platform Studies co-founder Nick Montfort to interpret a single line of code in BASIC was so provocative. In our first book-length publication

10 Mark C. Marino: Code as Ritualized Poetry: The Tactics of the Transborder Immigrant Tool. In: *Digital Humanities Quarterly* No. 1 (2013): <http://www.digitalhumanities.org/dhq/vol/7/1/000157/000157.html> (accessed September 24, 2023).

11 Evan Buswell: Comment, In Pursuit of Natural Language: FLOW-MATIC -- WORKBENCH. In: *CCS Working Group 2014* (2014), http://wg14.criticalcodestudies.com/discussion/comment/474#Comment_474 (accessed September 24, 2023).

of CCS, 10 authors explored `10 PRINT CHR$(205.5+RND(1));: GOTO 10`.¹² In the exploration of that eponymous line of code, artists, scholars, and programmers approached the code from every angle they could find, from the appeal of mazes to experimental dance, from the culture of early home computing to the practices of textile design. We read the tokens and wrote programs to understand this simple yet generative BASIC one-liner. Bringing our diverse disciplinary backgrounds to the interpretation, we examined the code through the lenses of art, sociology, anthropology, as well as creative computing. That endeavor was probably the clearest early demonstration of the fruitfulness of extended code exegesis.

What is code exegesis? I recently had a conversation with some seasoned computer programmers with whom I'm analyzing the code to Joseph Weizenbaum's *ELIZA* system on which ran one of the most discussed programs, *DOCTOR*. Published in 1966, *DOCTOR* (which is often conflated with *ELIZA*) plays the part of a Rogerian psychotherapist in conversational exchange.¹³ During our collective live code reading, we trudged line-by-line through the code written in a fairly niche language named *MAD-SLIP*. As we read on, one of the programmers, who possessed a deep knowledge of the *MAD-SLIP* code asked, "What are you trying to note?" In other words, what were we hunting for? I replied that we were engaged in exegesis, which I glossed as to walk out and about the code, starting, of course, from what we might call its denotative meaning and moving out to the connotative meaning. Such terms might be seen as basic practice for those in the interpretive arts, but for a programmer whose chief responsibility is to develop an almost intuitive fluency with the denotative meanings, so the process of signification becomes almost transparent, such a venture was not only unusual but almost antithetical to their daily encounter with code. If one stopped to

12 Nick Montfort et al. (ed.): *10 PRINT CHR\$(205.5+RND(1));: GOTO 10*. Cambridge MA: The MIT Press 2013.

13 Joseph Weizenbaum: *ELIZA – a Computer Program for the Study of Natural Language Communication Between Man and Machine*. In: *Commun. ACM* 9/1 (1966), 36–45. *ACM Digital Library*, <https://doi.org/10.1145/365153.365168>.

consider the significance of every token they used, they would not get very far. To partake in a leisurely exegesis would seem antithetical and impractical. Where might such an impractical stroll lead?

Our discoveries in *ELIZA* are still in their early stages, but allow me to offer two initial findings. One part of the code turns *ELIZA* into an editor of scripts. The *CHANGE* method, which Weizenbaum alludes to in his article, extends *ELIZA* from a platform for having conversations with a bot to a platform for changing the bot's script (for example, the *DOCTOR* script). Programming becomes conversational through this code. Another discovery led us to conclude that the *ELIZA* code Weizenbaum describes in the article is not the code that he used to generate the oft-quoted dialogue with *DOCTOR*. Additional discoveries will be forthcoming in our book manuscript, but I do want to mention that pouring through the *ELIZA* and *DOCTOR* code with this team of experts, with very diverse backgrounds and intellectual approaches, an exploration that began during a CCS Working Group, has once again reinforced my sense that critical code studies is best done in groups.

2. Mermaids and Pits

Critical code studies may have most to offer when supplementing other approaches. Jessica Pressman, Jeremy Douglass, and I demonstrated that in our book *Reading Project*, a collaborative exploration of one digital object.¹⁴ In that collective reading, critical code studies proved to be a useful tool to complement the other approaches such as close reading the text, historical research into the references of the art work, and visualizations drawing on the methods of cultural analytics. First, our exploration of the code of this distracting flashing piece helped us find the entirety of the story in the form of a very readable text file. However, in the same file, we found the series of subliminal words that flash

14 Jessica Pressman/Mark C. Marino/Jeremy Douglass: *Reading Project: A Collaborative Analysis of William Poundstone's Project for Tachistoscope {Bottomless Pit}*. Iowa: University of Iowa Press 2015.

upon the screen in a variable named "spam," which transformed how we interpreted them from the subliminal to the suboptimal, not that which alludes our notice, but that which tries to slip past our filters. As we examined the ActionScript code further we likewise discovered that the call to these show these spam words alternate with every story word and actually precede the call to the story words, giving them priority, showing their centrality or even primacy in the piece. Such explorations show not only the fruitfulness of CCS in explorations of works of art, but also how combining CCS with other practices can lead to stronger overall readings of works.

We demonstrated the usefulness of CCS again in our collective reading of *FISHNETSTOCKINGS*, an immersive work created by Joellyn Rock and a team of collaborators.¹⁵ In "Entanglements," co-written with Pressman and Diana Leong, we dove into the work looking at the legacy of mermaids, the aesthetics of silhouette art (featured throughout the piece), and of course the nature of the code, which led us not so much to fish as birds, since the flocking algorithm of the mermaids drew upon the code for bird flocking. As in our exploration of Poundstone's piece in *Reading Project*, this collaborative reading was enriched by the exploration of code, which showed us not only the way the artwork functioned but also revealed notes on its development and thematic tropes in names of objects and processes, revealing and resonating with themes in the work, such as hybridity. CCS does not have to be the sole focus in a reading, but now, as these methods have become more developed, they can become part of the toolset of the reader of digital objects in the digital humanities.

To read code is not merely to examine aesthetic objects. In Critical Code Studies Working Groups and in other collaborations, we have examined the Apollo Lunar Lander Code only to find "Burn Baby Burn" in the comments, a reference that tied the extraterrestrial moon shot to the very Earth-bound race riots of the 1960s in Watts, California. A group

15 Marc C. Marino/Diana Leong/Jessica Pressman: Entanglements. In: *The Digital Review* 2 (2022): https://thedigitalreview.com/issue02/marino_entanglements/index.html (accessed September 24, 2023).

based out of the Humanities and Critical Code Studies Lab, including Jeremy Douglass, Sarah Ciston, and Zach Mann and I have examined the code for predictive policing software to trace how it reinscribes racial and ethnic bias. Our work on racial justice, follows in the footsteps of Safiya Noble,¹⁶ Ruha Benjamin,¹⁷ and Joy Buolomwini (Algorithmic Justice League) as we formed an Anti-Racist Critical Code Studies Reading Group.

Recently, we have collected and published a group of early scholars in two special issues of *Digital Humanities Quarterly*. The first set of essays involve close readings of code for Alexa,¹⁸ an early BASIC game FTBALL,¹⁹ two artworks by Daniel Howe,²⁰ and another piece from the Ryan Veeder competition, a sonnet written in Inform 7.²¹ Scholars also attempt to apply CCS methods to artificial intelligence as in the work of Rita Raley and Minh Hua²² and of David Berry.²³ Examining the edges of pro-

16 Safiya Umoja Noble: *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York: NYU Press 2018.

17 Ruha Benjamin: *Race after Technology: Abolitionist Tools for the New Jim Code*. Cambridge: Polity 2019.

18 Lai-Tze Fan: Reverse Engineering the Gendered Design of Amazon's Alexa: Methods in Testing Closed-Source Code in Grey and Black Box Systems. In: *Digital Humanities Quarterly* 17/2 (2023).

19 Annette Vee: BASIC FTBALL and Computer Programming for All. In: *Digital Humanities Quarterly* 17/2 (2023).

20 John Cayley: Computational Art Explorations of Linguistic Possibility Spaces: Comparative Translingual Close Readings of Daniel C. Howe's Automatype and Radical of the Vertical Heart † . In: *Digital Humanities Quarterly* 17/2 (2023).

21 Jason Boyd: Poetry as Code as Interactive Fiction: Engaging Multiple Text-Based Literacies in Scarlet Portrait Parlor. In: *Digital Humanities Quarterly* 17/2 (2023).

22 Minh Hua/Rita Raley: How to Do Things with Deep Learning Code. In: *Digital Humanities Quarterly* 17/2 (2023).

23 David M. Berry: Tracing Toxicity Through Code: Towards a Method of Explainability and Interpretability in Software. In: *Digital Humanities Quarterly* 17/2 (2023).

NLP and Graph Theory,³² affect theory and sentiment analysis,³³ defactoring³⁴ and even reading code aloud.³⁵

The scholarship on code has also looked for new connections, epitomized by the new community Knit&Perl. Since the early days of critical code studies, we noticed a connection between coding and stitchcraft. It wasn't just the number of programmers who also engaged in some form of fibre art, such as sewing or crocheting, but in the nature of the two practices. Both involved iteration, emergent design patterns, and processes of creation. While the same might be said of many practices, the grid-like format of many stitching-related activities had so much in common with the grid of the computer screen, it was hard to ignore. Early on in our examination of 10 PRINT, I asked who else might have the knowledge of the kind of emergent pattern-making of this algorithm. The answer: textile workers. In primers on sewing, I found the algorithmic instructions that produced similar patterns, only minus the randomness.³⁶ In conversation with prominent digital humanists who also sewed, Anne Sullivan, Anastasia Salter, and I decided to form a community to discuss the intersections between stitchcraft and coding. So was born Knit&Perl, an online community dedicated to the discussion of programming and stitchcraft.

Much of the community has appeared in the Critical Code Studies Working Group, two special issues of *Digital Humanities Quarterly*, and strings of articles in *electronic book review*. These published readings present only the most preliminary of findings and demonstrations in a

32 Chris Tanasescu/Raluca Tanasescu: NLP and Graph-Theory-Based Coding for Text (Corpus) Analysis. A Comparative Poetry and Philosophy Implementation Case Study. In: *Digital Humanities Quarterly* 17/3 (2023).

33 Jeffrey Moro: "Machine Reading for Atmosphere: Affect Theory and Sentiment Analysis in TextBlob. In: *Digital Humanities Quarterly* 17/3 (2023).

34 Matthew Burton/Joris Van Zundert: Defactoring the Pace of Change. In: *Digital Humanities Quarterly* 17/3 (2023).

35 Mace Ojala/Katrine Kjær: Reading Code Aloud. In: *Digital Humanities Quarterly* 17/3 (2023).

36 Mark C. Marino: The ppg256 Perl Primer: The Poetry of Techneculture. In: *Emerging Language Practices* 1 (2012).

field too new to have fixed methods. They are developed in every new code reading.

"The end of 'The Gay Science' offers three plaintive lines in the comments:

[I hope you can read this.]

[I hope this works.]

[I hope.]"

Every programmer can relate to the hope that their code functions properly; however, the fictional programming persona writing the code for *The Gay Science* seems to hang their hopes on a reader who receives the hidden message placed inside the code. The first years of critical code studies suggest that there are many messages to be gleaned, messages full of meaning that we discover and create by looking deeply into the source code. In code, we find intercepted messages to a machine that have even more meaning for us.

Autor:innen

Martin Bartelmuß ist Postdoc am Institut für Germanistik der Heinrich-Heine-Universität Düsseldorf mit den Forschungsschwerpunkten French Theory, Object-Oriented Ontology, Animal und Plant Studies sowie Materialität, Medialität und Schriftlichkeit.

Lucas Falkenhain studierte Germanistik und Kunstgeschichte und befindet sich derzeit im Master der interdisziplinären Medienwissenschaft. Seit 2018 ist er Mitarbeiter an der fachübergreifenden Einrichtung »Studierendenakademie« der Heinrich-Heine-Universität Düsseldorf. Er beschäftigt sich mit den kulturtechnischen Implikationen automatisierter Text- und Bildtechnologien.

Gabriele Gramelsberger hat den Lehrstuhl für Wissenschaftstheorie und Technikphilosophie an der RWTH Aachen inne. Ihr Forschungsthema ist die Transformation der Wissenschaft durch die Digitalisierung, insbesondere durch die Einführung neuer Forschungsmethoden wie der Computersimulation und des maschinellen Lernens. Gemeinsam mit Markus Rautzenberg war sie in dem Forschungsprojekt »Mind the Game!« aktiv. 2018 gründete sie mit Unterstützung des Stifterverbandes das Computational Science Studies Lab an der RWTH Aachen. Sie ist Mitglied der Nordrhein-Westfälischen Akademie der Wissenschaft und der Künste sowie Direktorin des Käte Hamburger Kollegs c:o/re »Kulturen des Forschens«.

Joachim Harst war von 2018 bis 2024 Juniorprofessor für Komparatistik an der Universität zu Köln und vertritt ab dem Sommersemester 2024 die Professur für Allgemeine und Vergleichende Literaturwissenschaft an der Universität des Saarlandes. Zu seinen Forschungsschwerpunkten zählen: Investigative Kunstformen; Mediengeschichte der Psychoanalyse; Verbindlichkeit in Recht, Religion und Literatur; sowie Poetik und Rhetorik von Trauerspiel und Tragödie. Zuletzt erschienen: »Virtuelle Investigationen. Transformationen des Indizienparadigmas zwischen Sherlock Holmes und Forensic Architecture«, *Medienkomparatistik* (2023), 23–44; »Machine Analysis. Westworld and the Media History of Psychoanalysis«, in: *Therapie der Dinge? Materialität der Psychoanalyse in Literatur und den bildenden Künsten*, hg. von Martin Bartelms und Friederike Danebrock, Bielefeld: transcript, 2023, 211–224; »Universalgeschichte des Ehebruchs.« *Verbindlichkeit zwischen Literatur, Recht und Religion*. Göttingen: Wallstein, 2021.

Stefan Höltgen (Dr. phil., Dr. rer. nat.) hat von 1995 bis 2000 Germanistik, Philosophie, Soziologie und Medienwissenschaft in Jena studiert. 2009 wurde er in Neuer Deutscher Literaturwissenschaft an der Universität Bonn promoviert. Zwischen 2011 und 2022 war er wissenschaftlicher Mitarbeiter an der Medienwissenschaft der Berliner Humboldt-Universität, wo er die Archäologie früher Mikrocomputer und ihre Programmierung erforscht hat. Zu diesem Thema wurde er sich 2020 im Fach Informatik promoviert. Seit 2022 betreibt er an der Universität Bonn ein DFG-Forschungsprojekt zur Computerphilologie, in dem er die technischen Kulturen der BASIC-Programmierung untersucht. Er ist Mitherausgeber der Buchreihe »Computerarchäologie«, Herausgeber der Lehrbuchreihe »Medientechnisches Wissen« und Autor zahlreicher Schriften zur Geschichte und Theorie digitaler Medien. Informationen: www.stefan-hoeltgen.de

Lore Knapp ist Akademische Rätin auf Zeit im Fach Literaturwissenschaft an der Universität Bielefeld. Sie studierte Neuere deutsche Literatur, Theaterwissenschaft und Musikwissenschaft an der Freien Universität Berlin sowie Musikpädagogik mit Hauptfach Violoncello an der

Universität der Künste Berlin. Sie absolvierte ihr Promotionsstudium an der Friedrich Schlegel Graduiertenschule für literaturwissenschaftliche Studien der Freien Universität Berlin sowie am Gonville and Caius College der Universität Cambridge. 2017 wurde sie in das Junge Kolleg der Nordrhein-Westfälischen Akademie der Wissenschaften und der Künste aufgenommen. 2021 erwarb sie eine Venia Legendi in den Fächern Neuere deutsche Literatur und Allgemeine und vergleichende Literaturwissenschaft. Sie forscht zur Geschichte der Ästhetik, zum Einfluss der Digitalisierung auf literarische Schreibprozesse, zu Formen des Kunstreligiösen, zu Verbindungen von Empirismus und Ästhetik sowie zur Rezeptionsgeschichte John Lockes.

Mark C. Marino is a Professor (Teaching) of Writing at the University of Southern California, where he directs the Humanities and Critical Code Studies Lab where he is a Generative AI Fellow. Since 2008, he has been the Director of Communication of the Electronic Literature Organization (<https://eliterature.org>). His works include »Living Will,« »a show of hands,« and »Marginalia in the Library of Babel.« He was one of ten co-authors of *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10* (<https://10print.org>) (2013) and was a collaborator with Jessica Pressman and Jeremy Douglass on *Reading Project: A Collaborative Analysis of William Poundstone's Project for Tachistoscope {Bottomless Pit}* (2015). His latest books are *Critical Code Studies* (2020) <https://criticalcodes.com> and *Hallucinate This! an authorized Autobotography of ChatGPT* (2023) <https://bit.ly/halthis>.

Julia Nantke ist seit 2019 Juniorprofessorin (mit Tenure Track) für Neuere deutsche Literaturwissenschaft mit dem Schwerpunkt Digital Humanities für Schriftartefakte an der Universität Hamburg. Forschungsschwerpunkte: Digitale Literatur und digitale Literaturwissenschaft, Literaturtheorie, Materialität und Medialität von Literatur, Editionswissenschaft, Literatur um 1900. Aktuelle Forschungsprojekte zur KI-gestützten Erschließung von Handschriften, zur computationalen Analyse von Textähnlichkeiten und zum Umgang mit Born digital-Nachlässen. 2024 erscheint *Autor:innenschaft und Arbeit. Zum Verhältnis*

von Praktiken, Inszenierung und Infrastrukturen. Hg. v. Julia Nantke und Alena Heinritz.

Alexander Nebrig ist Professor für Neuere Deutsche Literaturwissenschaft an der Heinrich-Heine-Universität Düsseldorf. Zu seinen aktuellen Forschungsgebieten zählen die Geschichte und Theorie der Schrift, die Beziehung der Literatur zur Ethik und die Geschichte des interlingualen Lizenzraumes.

Tilo Reifenstein ist senior lecturer in Critical Studies an der York St John University (UK). Der Beitrag in diesem Band wurde durch eine International Fellowship am Kulturwissenschaftlichen Institut Essen (KWI) gefördert.

Claus-Michael Schlesinger ist Literatur- und Kulturwissenschaftler und beschäftigt sich mit den Verhältnissen von Technik und Ästhetik in Geschichte und Gegenwart. Er arbeitet derzeit als Mitarbeiter an der Universitätsbibliothek der Humboldt-Universität zu Berlin. Schwerpunkte seiner Arbeit sind Digitale Literatur und ihre Archivierung, Geschichte und Theorie der Informationsästhetik, Geschichte der Meteorologie und Klimatologie, Methoden und Infrastruktur in den Digital Humanities.

Philipp Schönthaler ist freier Schriftsteller, seine Bücher erscheinen bei Matthes & Seitz Berlin. Zuletzt: *Wie rationale Maschinen romantisch wurden. KI, Kreativität und algorithmische Postrationalität* (2024), *Die Automatisierung des Schreibens & Gegenprogramme der Literatur* (2021) sowie der Roman *Der Weg aller Wellen* (2019).

Christian Schulz (Dr.), ist Nachwuchsgruppenleiter und assoziiert mit dem Arbeitsbereich Kulturen der Digitalität am Institut für Medienwissenschaften an der Universität Paderborn sowie dem Transregio 318 »Constructing Explainability«. Seine Forschungsschwerpunkte sind soziale Medien und ihre Medientheorien, Algorithmen und KI, Datenpraktiken, Theorien des Subjekts und digitale Fotografie. Aktuelle Veröffentlichungen: *Infrastrukturen der Anerkennung. Eine Theorie*

sozialer Medienplattformen (2023), Frankfurt a.M./NewYork: Campus;
A New Algorithmic Imaginary (2023), in: *Media, Culture & Society* 45(3),
646–655; Reziprozität und das alteritäre Dritte – über die wechselsei-
tigen Verflechtungen von PageRank und sozialen Medien (2022), in:
Behemoth – A Journal on Civilisation 15(2).

