

.exe Software und ihre Bedeutung für
eine erziehungswissenschaftliche
Medienforschung

Dan Verständig

Anhand der Abkürzung .exe (executable) untersucht der Beitrag die Komplexität hinter der Ausführung von Programmen hinsichtlich technischer Abhängigkeiten aber auch sozialer und kultureller Bedingtheiten. Damit ergeben sich forschungsbezogene Konsequenzen, die im Beitrag über die Beziehung zwischen den Software Studies und einer erziehungswissenschaftlichen Medienforschung diskutiert werden.

Es hat heute fast einen Hauch von Ironie, dass Kittler 1992 für ein Symposium im Silicon Valley seine These postulierte, dass es keine Software gäbe – denn fast zeitgleich stellte Microsoft die Version 3.1 seines Betriebssystems Windows mit grafischem User Interface vor, welches nicht zuletzt aufgrund seiner Oberfläche und der damit verbundenen Abstraktion zur Hardware eine weltweite Erfolgsgeschichte feierte. Programme, die auf dem Betriebssystem Windows ausgeführt werden, sind sogenannte Executables, ausführbare Programmdateien, die auf die Endung **.exe** enden. Auch wenn es sich um eine sehr bekannte Abkürzung handelt, ist die Ausführung (execution) von Programmen längst nicht auf diese Formate beschränkt: Es können ganz unterschiedliche Skripte und Kommandos wie Binary Executables (.bin) oder andere Befehle (siehe **cmd**, Breiter 2023) sein, die auf einem bestimmten Betriebssystem ausgeführt werden. Software ist damit immer auch in Software eingebettet. Die Universalität des Computers hat zur Folge, dass wir nicht mehr sehen, ob einfach Zeichen gesetzt werden oder aber der Befehl gegeben wird, Zeichen zu setzen. Doch neben dieser Form der Verschleierung war für Kittler (1992) vor allem eines wichtig: Software wird immer auf Hardware ausgeführt und ist untrennbar mit ihr und ihrer Materialität verbunden.

Software zählt. Heute womöglich mehr denn je. Wem verdankt der Computer als universelle Rechenmaschine seine magische Kraft der Unverwundlichkeit? Der Software. In welcher Form nehmen die meisten modernen Anwendungen und Apps Gestalt an und werden daher von ihren Benutzer:innen wahrgenommen? Über digitale Medien. Software hat inzwischen eine Vielzahl von physischen, mechanischen und elektronischen Technologien ersetzt, die vor dem 21. Jahrhundert verwendet wurden, um kulturelle Artefakte zu erstellen, zu speichern, zu verteilen und mit ihnen zu interagieren. Software kann, mit Manovich (2013) gesprochen, als Schnittstelle zur Welt, zu anderen, zu unserem Gedächtnis und unserer Vorstellungskraft beschrieben werden. Software macht aus Smartphones ein Schweizer Taschenmesser in der digitalen Welt. Das Denken über Software selbst ist über weite Strecken ihrer Geschichte gemeinhin technisch geblieben. Dies ändert sich durch

verschiedene Strömungen in Kunst, Kultur, Gesellschaft und Wissenschaft. Dementsprechend ist eine weite Auslegung von Software heute grundlegend für historische und zeitgenössische Analysen, die sich im Schnittfeld von Technik-, Medien- und Kulturgeschichte bewegen. Software takes command – und das in mindestens zweierlei Hinsicht: Software ist allgegenwärtig und zugleich immer auch von den Eingaben der Menschen abhängig, die mit der Software arbeiten, die Daten für den jeweiligen Input produzieren. Software braucht einerseits Befehle, andererseits bestimmt sie öffentliche, wissenschaftliche, kulturelle und politische Diskurse und durchzieht das Private in einem bislang unbekannten Ausmaß (Pritchard/Snodgrass/Tyzlik-Carver 2018).

Insbesondere die Hinwendung zu Software hat damit auch im Wissenschaftsbetrieb einige Spuren hinterlassen und fordert nicht nur in der gegenstandsbezogenen Auseinandersetzung dazu auf, sich zu ihr zu verhalten. Will man sich in der Erziehungswissenschaft den beschriebenen Phänomenen nicht verschließen, dann scheint es nur folgerichtig, auch nach den Produktionsbedingungen von digitalen Medien zu fragen und Designprinzipien in den Blick zu nehmen. Das bedeutet, dass man sich mit den kulturellen, sozialen, aber auch politischen Einschreibungen von Software befasst, um deren implizite Logiken und explizite Wirkweisen analytisch erfassen zu können. Dabei weist die Ausführung einer Softwareanwendung viele Ähnlichkeiten zur Ausführung von Forschungsprogrammen auf: So ist die Ausführung eines Prozesses oder Programms nicht einfach die reibungslose Erledigung einer Aufgabe, vielmehr kann sie als ein multidimensionales Wechselverhältnis von Befehl und Kontrolle, von Eingabe und Ausgabe, von Daten und – inzwischen vorrangig digitaler/disziplinärer – Infrastruktur verstanden werden. Der Prozess der Ausführung ist damit nur ein Bestandteil von vielen und eingebettet in eine umfangreiche soziotechnische Struktur; ebenso wie der Prozess des Erkenntnisgewinns in den Humanwissenschaften.

In drei Schritten wird im Folgenden eben dieses Wechselverhältnis herausgearbeitet und anschaulich gemacht. Erstens wird die stetig steigende Komplexität und Kontingenz sowohl in gesellschaftlichen als auch wissenschaftlichen Kontexten betrachtet und im Hinblick auf die Software Studies verortet (*executing complexity*). Deren konsolidierenden Bewegungen sind von einer spezifischen *transdisziplinären* Qualität geprägt, die nicht zuletzt für eine erziehungswissenschaftliche Medienforschung bedeutend ist, wie im zweiten Schritt gezeigt wird (*executing transdisciplinarity*). Worum es dabei letztend-

lich geht, sind disziplinäre Suchbewegungen, das Herstellen von Ordnungen und das Durchkreuzen bestehender Ordnungen sowie Denkweisen und der Umgang mit Wissen im Horizont der Digitalität. Der Beitrag schließt daher mit einer kursorischen Abhandlung über *executing order*.

Executing Complexity

Der Rechtswissenschaftler Lessig (1999) hat die Form der Komplexität über die regulative Macht des Codes in seinem Band *Code is law* festgehalten. Er beschreibt darin eine gewissermaßen neue Form der Regulierung, bei der (rechnerbasierte und digital vernetzte) Technologien eingesetzt werden, um bestehende Regeln durchzusetzen. Die Einschreibung von Werten in Hard- und Software haben bei Lessig unmittelbare Auswirkungen auf das gesellschaftliche Miteinander, sie seien konstitutiv für die Erfahrung von Welt. Unter dem Begriff des Cyberspace – damals en vogue, heute eher anachronistisch – fokussiert er sich auf die Architektur, die durch Hardware, aber auch durch Software definiert sei und die er mit *Code* zusammenfasst. Diese Architektur biete bestimmte Prinzipien ein; sie lege die Bedingungen fest, unter denen Menschen den Cyberspace nutzen. Die Formalstruktur des Raumes definiert also, was in ihm möglich ist und was nicht – und diese Bedingungen beziehungsweise Möglichkeiten beeinflussen wiederum Innovationen in diesem Raum: Einige Architekturen laden zur Innovation ein, andere hemmen sie. Es wird nicht zuletzt durch spätere Arbeiten deutlich (Lessig 2010; van Schewick 2010), dass jene Einschreibungen weitreichende Konsequenzen – nicht nur für die Innovationskraft des Netzes, sondern für die Konstitution gesellschaftlicher Zusammenhänge insgesamt – haben.

Im Zeitraum von 2004 bis 2012 haben sich auch verschiedene Ansätze entwickelt, die Code und Software vor dem Hintergrund ihrer kulturellen Wirkweisen und Einbettungen diskutieren und damit auch gesellschaftstheoretische Bezüge eröffnen. Mit dem Fokus auf Code im Sinne von Quellcode lassen sich hier insbesondere die Arbeiten von Adrian MacKenzie (2005), Florian Cramer (2005) und Alexander Galloway (2004) nennen, die durch ihre Aufsätze und Buchprojekte eine grundlegende Öffnungsbewegung bewirkt haben. Diese sind stark von den Cultural Studies geprägt. Anschließend daran und fast zeitgleich haben sich die Software Studies (Fuller 2008; Wardrip-Fruin 2012; Manovich 2013) und Platform Studies (Montfort und Bogost 2009) mit jeweiligen Buchreihen bei MIT Press herausgebildet, wie Marino (2020: 19) in seinem Einführungsband über die Critical Code Studies zusammenfasst. Die Software Studies nehmen aus ganz unterschiedlichen Perspektiven

und mit verschiedenen Zielsetzungen die Herausforderungen in den Blick, die sich durch, mit und über Software ergeben. Zwar wurde die Implementierung und Nutzung von Software zuvor schon in Bereichen wie den Internet Studies oder New Media Studies untersucht, jedoch meist als Begleiterscheinung, die es zu adressieren galt, um die sozialen, kulturellen oder politischen Phänomene der jeweiligen Perspektiven zu beleuchten. Mit dem Aufkommen der Software Studies wurde die Software schließlich zum eigenständigen Untersuchungsgegenstand. Matthew Fuller (2008) beschreibt das Projekt der Software Studies im Eröffnungsband zur gleichnamigen Buchreihe als eine Sammlung von Kurzstudien, die spekulative, erklärende und kritische Texte über spezifische digitale Artefakte, Sprachen und logische Strukturen zusammenbringt. In insgesamt 39 Beiträgen werden zentrale Begriffe zu Software in Anlehnung an ein Lexikon explorativ eingeführt, um das Verständnis von Software als *immaterielles* Artefakt zu überwinden (ebd.: 4). Dabei dienen nicht nur kultur- und medienwissenschaftliche Zugänge als Impulsgeber, sondern auch Beiträge aus den Science and Technology Studies. Software wird als etwas verhandelt, das eine eigene Geschichte hat und somit nicht nur den harten, weil technologischen, Gesetzen unterliegt. Software ist vielmehr ein kontingentes Produkt, welches durch Medientechnologie ebenso bedingt ist wie durch soziale, institutionelle und kulturelle Rahmenbedingungen. Neben den theoretisch-konzeptionellen Beiträgen, die sich in erster Linie an der Frage nach der Bedeutung von Code und Software abarbeiten, ergeben sich in diesem Zusammenhang auch methodische und methodologische Herausforderungen und Fragestellungen. Letztere zielen einerseits unter Einfluss der digitalen Geisteswissenschaften auf computergestützte Analysen ab, andererseits berücksichtigen sie auch hermeneutische Ansätze.

Daneben ist die Forschung mit und über Software allein aufgrund ihrer Beschaffenheit schwierig, wie Chun (2011: 3) festhält. Denn Software in historischer Perspektive zu beforschen, birgt die Herausforderung, dass alte Programme auf aktueller Hardware und aktuellen Betriebssystemen womöglich gar nicht mehr ausgeführt werden können. Selbst der Zugang zu alter Software ist grundlegend problematisch. Gleichzeitig scheint eine kategorische Trennung von Code und Software als Engführung, die der Charakteristik von Software und ihrer performativen Kraft selbst widerspricht. Chun (2011) verdeutlicht diesen Punkt entlang der Diskussion um Source Code, Software und die spekulativen Räume, die über ihre Anordnungen, Ausführungen und Bedeutungszuweisungen entstehen: „Code does not always or automatically do what it says, but it does so in a crafty,

speculative manner in which meaning and action are both created“ (ebd.: 24). Dementsprechend fasst er Source Code eher als *re-source* statt als Quelle:

„Source code is more accurately a re-source, rather than a source. Source code becomes the source of an action only after it — or more precisely its executable substitute — expands to include software libraries, after its executable version merges with code burned into silicon chips; and after all these signals are carefully monitored, timed and rectified. Source code becomes a source only through its destruction, through its simultaneous nonpresence and presence.“ (Ebd.: 24f.)

Die Komplexität bei der Betrachtung von Software lässt sich also nur schwer von der Unbestimmtheit trennen, die Source Code schon hervorbringt. Gleichzeitig ist Source Code nicht die alleinige Ressource zur tatsächlichen Ausführung von Prozessen und Abläufen, denn Software ist kein immaterielles, stabiles oder gar wertneutrales Produkt. Vielmehr handelt es sich bei der Anwendung, Ausführung und Produktion von Software um ein vielschichtiges und wandelbares Beziehungsgeflecht, das geprägt ist von verschiedenen diskursiven, ökonomischen und materiellen Praktiken (Kitchin/Dodge 2011: 37), die Kontingenz hervorbringen und Komplexität steigern (Verständig 2020). Das Ergebnis dieser Kontingenz ist die steigende Notwendigkeit einer inter- und transdisziplinären Befragung von digitalen Architekturen auf ihre funktionalen und ethischen Grenzen sowie Reichweiten hin. Eine erziehungswissenschaftliche Medienforschung kann sich diesen – den Medien gewissermaßen vorgelegerten – Prozessen in besonderer Form kritisch-reflexiv zuwenden, um bisherige Herausforderungen in pädagogischen Handlungsfeldern aber auch in der Theoriebildung angemessen zu bearbeiten und neue Problemstellungen überhaupt erst sichtbar zu machen. Erziehungswissenschaftliche Fragestellungen entlang von medialen Gegenstandsbereichen zu diskutieren, erfordert auch eine methodische Offenheit, weswegen eine erziehungswissenschaftliche Medienforschung schon per se Komplexität aufweist und ausführt, etwa indem sie auf disziplinübergreifende Zusammenhänge verweist (*executing complexity*). Doch wie überhaupt versteht sich eine erziehungswissenschaftliche Medienforschung als wissenschaftliche Disziplin? Diese Frage beantwortet Schäffer (2012: 137), indem er einen grundlegenden Überblick gibt:

„Eine erziehungswissenschaftliche Medienforschung sollte Erziehungswissenschaft als sozial- und kulturwissenschaftlich informierte sowie bildungs- und erziehungstheoretisch re-

flektierte Disziplin auffassen, die sich nicht nur Erziehungsprozessen zwischen ‚Erziehendem‘ und ‚Zögling‘ im engeren Sinne widmet, sondern auch solche des Lernens, der Sozialisation und der Bildung aller Altersgruppen in den unterschiedlichsten Kontexten in den Blick nimmt. Eine hieran orientierte erziehungswissenschaftliche Medienforschung umschließt insofern im Grunde alle Formen und Modalitäten des lernenden, bildenden, sozialisierenden und erziehenden Umgangs mit Medien(technologien) sowie auch dessen historische, soziale, kulturelle und gesellschaftliche Voraussetzungen.“

Eine erziehungswissenschaftliche Medienforschung kann mit Blick auf Software (und Code) auch über die Betrachtung von medialen Praktiken im Anschluss an Technologien hinausgehen. Denn bereits mit dem *Schreiben* von Code und damit der *Produktion* von Software werden Abhängigkeiten deutlich, die sich nicht nur auf rein technische Zusammenhänge reduzieren lassen, sondern immer auch soziale und kulturelle Ausdrucksformen in sich vereinen, wie am Beispiel einer einzigen Zeile Code durch Montfort/Baudoin/Bell et al. (2014) gezeigt wurde. Gleichzeitig werden hier die Herausforderungen einer inter- beziehungsweise transdisziplinären Forschung besonders offensichtlich, denn auch die erziehungswissenschaftliche fundierte Betrachtung von Code und Software erfordert eine disziplinäre Offenheit insofern, als dass Lernen, Sozialisation, Bildung und Erziehung zumindest adäquat zu digitalen Technologien ins Verhältnis gesetzt werden können.

Executing Transdisciplinarity

Wissenschaft selbst ist unter den Bedingungen der Digitalität maßgeblichen Veränderungsprozessen ausgesetzt. Nicht nur, dass Wissensbestände einem beschleunigten Wachstum unterliegen und sich somit eine unüberschaubare Partikularisierung abzeichnet, auch die Halbwertszeit von Erkenntnissen ist davon in institutioneller, aber auch organisatorischer Hinsicht betroffen, wie Mittelstraß (2005: 18) hervorhebt. Fachliche und disziplinäre Grenzen, wenn man sie überhaupt noch wahrnehmen kann, drohen aufzubrechen; eine Entwicklung, die sich in voller Dynamik an der Entstehung um die Software Studies mitverfolgen lässt. Verschiedene Arbeiten der letzten Jahre haben gezeigt, dass diese sehr junge Disziplin in der Lage ist, alle geisteswissenschaftlichen Disziplinen in ihrer historisch gewachsenen Tradition mit beeindruckender Agilität auf sich zu beziehen und in ein spezifisches Verständnis im Umgang mit Code, Software, Algorithmen und Daten zu überführen.

Wenngleich die Bestimmung der Software Studies eine Konsolidierungsphase nahelegt, dann ist doch mit den verschiedenen interdisziplinären Suchbewegungen, die dieser Konsolidierung vorausgingen, eine hochgradige Vielfalt in den Perspektiven eingeschrieben, die ihren historisch gewachsenen Kern in den jeweils unterschiedlichen Disziplinen haben. Folgt man Mittelstraß, so scheinen Bestrebungen, diesen Entwicklungen über den Begriff der Interdisziplinarität im Sinne einer „Reparaturvorstellung“ (ebd.) entgegenzutreten, nur bedingt angemessen zu sein. Mittelstraß adressiert demgegenüber den Begriff der Transdisziplinarität: Diese bleibe zwar entlang ihrer argumentativen Entstehung und der dabei unterscheidbaren Stufen im Produktionsprozess nach wie vor auf disziplinäre Kompetenzen angewiesen, werde jedoch nicht nur auf disziplinäre Gegenstände angewendet und konstituiere damit schließlich eine neue *Disziplinarität*. Die Disziplinen blieben in diesem Prozess nicht das, was sie waren, sie veränderten sich selbst – gewissermaßen im Fluss der Aushandlung – mit den Gegenständen, Theorien, Methoden und Methodologien, die sie prägten. Transdisziplinarität ist damit nicht nur als ein Forschungs- und Arbeitsprogramm zu sehen, das sich sowohl auf wissenschaftlichen Erkenntnisgewinn als auch auf die außerwissenschaftliche Lösung gesellschaftlicher, kultureller oder ökologischer Herausforderungen bezieht; durch Transdisziplinarität wird Wissenschaft im Kern adressiert und die Ordnung des wissenschaftlichen Wissens zur Disposition gestellt sowie auf seine jeweiligen Grenzen hin befragt. *Executing transdisciplinarity* heißt dann auch, dass sich eine bestimmte Kultur von Wissenschaft gegenüber anderen vor dem Hintergrund der Ausdifferenzierung von Wissensbeständen durchzusetzen vermag. Die Einarbeitung des transdisziplinären Wissens in unterschiedliche Diskurstraditionen und das Schaffen von anschlussfähigen Passungen ist dann in besonderem Maße von der Offenheit oder der Öffnung der jeweiligen argumentativen Sichtweisen und Bestrebungen abhängig. Der von Mittelstraß (2005) als bedrohlich eingestuft Unübersichtlichkeit liegt ein Hang zum Verlust der eigenen disziplinären Identität zu Grunde. Diese Problematik lässt sich beispielhaft am deutschsprachigen Diskurs über Bildung und digitale Medien nachzeichnen. Neben Bestrebungen zur interdisziplinären Zusammenarbeit, wie etwa zwischen Medienpädagogik und Informatik, ergeben sich immer wieder auch Abgrenzungstendenzen. Während Positionspapiere wie die *Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt* (2016) einen Ansatz zur Kompromissbildung darstellen, sind es Ergänzungen wie das *Frankfurt-Dreieck zur Bildung in der digital vernetzten Welt* (2019), die durch die

konsensuale Reduktion von Begriffen und Inhalten erneut Komplexität aufbauen. Zweifelsohne herrscht Konsens über die Bedeutung digitaler Technologien für die Herstellung individueller Orientierung und für die Gestaltung überindividueller Rahmen (wie institutioneller Bildungssettings) sowie für die damit verbundene Organisationsentwicklung. Dennoch zeichnen sich durch begrifflich geprägte Geltungsansprüche Herausforderungen in der Ausführbarkeit des verabschiedeten Programms *Transdisziplinarität* ab. Die Differenz der Diskurslinien wird auch dann deutlich, wenn die Komplexität digital vernetzter Architekturen und ihre kulturell-historisch gewachsene Einbettung in diverse Kontexte – samt der sich damit verändernden Kontroll- und Machtverhältnisse (Galloway/Thacker 2007; Jörissen/Verständig 2017) – auf Deutungen der Kybernetik reduziert werden (Swertz 2020: 95). Die Differenz der Diskurslinien wäre eine Negation der „Gemachtheit von Daten“ (Dander 2016) und damit das Ausblenden der immanenten Einschreibungen von Werten und Möglichkeitsräumen in digitale Technologien. Demgegenüber stehen Diskurslinien der Informatik beziehungsweise Medien- und Kulturwissenschaft, die sich Problemen wie dem eines emergenten Bias (Friedman/Nissenbaum: 1996), der kulturellen Logik von Berechnungen (Golumbia 2009) und den Produktionsbedingungen von Software nähern (Cox/McLean 2013; Wachter-Boettcher 2017). Dementsprechend bietet die Rekonstruktion der Bewegungen innerhalb und nah an den Software Studies das Potenzial, sich vergleichbaren, über die eigene Disziplin hinausgehenden Problemstellungen zu widmen. Erinnern wir uns an den Einstieg in diesen Beitrag: Selbst wenn Kittlers These, dass es keine Software gäbe, mit den Software Studies gewissermaßen revidiert wurde, bleibt die Anerkennung dieser Perspektive doch ein fester Bestandteil in eben jenen aktuellen Auseinandersetzungen: Etwa wenn ihr von Marino (2020: 161ff.) ein ganzes Kapitel gewidmet wird oder sie bei Cox und McLean (2013) Erwähnung findet, um den Prozess des Verstehens, die Logik des Denkens über Software und Hardware, zu entschlüsseln. Das Denken über Software prägt damit die Ausführungen von Forschungsansätzen, die sich aus verschiedenen disziplinären Perspektiven zusammensetzen und dadurch erst neue Impulse ermöglichen. Die Software Studies sind losgelöst von einem Denkschema. Sie sind keine feste Disziplin, eher ein intellektuelles Paradigma, das sich darüber auszeichnet, dass ihre Vertreter:innen an der Erforschung der Beziehung zwischen Computer und Medien interessiert sind. Auch inhaltlich lassen sich die Software Studies durch ihr wesentliches Merkmal der Offenheit kennzeichnen: Es muss nicht explizit um Software gehen, die eine funktionalistische

Zweckbindung oder der Medienproduktion gewidmet ist. Ebenso können auch literarische wie künstlerisch-ästhetische Ansätze behandelt oder mediale Taktiken diskutiert werden, etwa wie Menschen einer Überwachung auf der Straße entgegen oder wie der Schaffensprozess von Kunstwerken durch Software beeinflusst wird.

Über die Auseinandersetzung mit Software können also Räume für eine transdisziplinäre Zusammenarbeit entstehen, die Akteur:innen aus Wissenschaft und Gesellschaft – etwa über Citizen Labs oder andere partizipative Projekte – zusammenbringt. Software ist damit ein verbindendes Glied unterschiedlicher wissenschaftlicher Zugänge einerseits und von Akteur:innen unterschiedlicher Felder andererseits. In einer solchen Perspektivsetzung wird die Überwindung der eigenen disziplinären Grenzen im Sinne von innovativer Wissensproduktion verstanden. Es handelt sich jedoch um eine komplexe Entwicklung, die nicht ohne den Verlust bestehender Ordnungen und Traditionen vonstatten geht und damit längst keine tatsächlichen Innovationen im Wissensbetrieb garantiert.

Executing Order

Indem Wissenschaften über und durch digitale Medien sowie deren Technologien ausgeführt oder zumindest berührt werden, finden sich auch in den Medien selbst soziale sowie habituelle Einschreibungen. Es ist daher keine Überraschung, dass rund um die Software Studies und benachbarten Disziplinen stets kontrovers und kritisch über die Einschreibungen von Werten in Software, Daten und Code diskutiert wird. Die kritischen Positionen hierzu bewegen sich in einem breiten Spektrum von einer theoretisch-konzeptionellen Perspektive auf Macht- und Kontrollfragen (Galloway 2004; Chun 2011) über eine Betrachtung der ideologischen Konsequenzen von Software und Code (Galloway 2006; Golumbia 2009; Cox/McLean 2013; Soon/Cox 2020; Marino 2020) hin zu kritischen Ansätzen, die sich umfassend mit der Problematik um Bias und Diskriminierung (D'Ignazio/Klein 2020; McIlwain 2020) auseinandersetzen. Die unterschiedlichen Positionen proklamieren teils explizit, teils implizit eine Perspektive auf Software und Code, die auch für andere disziplinäre Zugänge relevant ist: Gewissermaßen grundlegend werden Beschreibungen des Menschseins und der Menschwerdung unter den Bedingungen der Digitalität beforcht und bestehende Ordnungen in Frage gestellt. Für eine erziehungswissenschaftliche Medienforschung bieten diese Impulse – wie die Problematisierung von Macht- und Kontrollfragen, Bias und Diskriminierung – geradezu ideale

Anknüpfungspotenziale, um Zugangs- und Teilhabefragen zu/an digitalen Technologien sowie praxeologische Konsequenzen im Umgang mit Software und Code einzuholen. Auf diese Weise kann es einer erziehungswissenschaftlichen Medienforschung gelingen, zu einem disziplinübergreifenden Erkenntnisgewinn nicht nur in der Theoriebildung, sondern auch in empirisch-analytischen Vorhaben zur Beantwortung fachimmanenter Fragestellungen zu kommen.

Executing order meint dann zum einen das Verflechten und Durchkreuzen von Ordnungen im *doing science*, zum anderen aber auch die reflexive Auseinandersetzung mit den Gegenständen, die zum Ausgangspunkt der Diskurse werden. In Anlehnung an Foucault (1981: 71) lassen sich Diskurse eben nicht bloß als ein Sprechen über Dinge charakterisieren, sondern eher als Praktiken, die systematisch jene Gegenstände hervorbringen, von denen sie sprechen. Dieser doppelten Differenz ist auch die subjektivierende Kraft von Software eingeschrieben, denn die Haltungen, die hinter den Praktiken mit, über und durch Software stehen, prägen Wissenschaft ganz grundlegend. Das reicht von der Ebene der Textproduktion, über die Untersuchung von Fragestellungen im Kontext der Digitalität, hin zur Forschung mit digitalen Methoden und den daran gebundenen forschungsethischen Herausforderungen. Die Bedeutung eines Texts, eines Arguments, eines Sprechakts, einer Artikulation hängt und hing immer von den Bedingungen ab, unter denen die jeweilige Artikulation entsteht und wie sie innerhalb eines Feldes oder disziplinären Diskurses überhaupt existiert. Kurzum: Die Ordnung des Diskurses wird nicht nur gesellschaftlich, sondern vor allem auch wissenschaftlich durch Prozeduren, Rituale und verschiedene Kräfte zugleich kontrolliert, selektiert und kanalisiert (Foucault 1991: 10f.). Digitale Technologien verändern damit nicht nur die Art und Weise, wie wir mit den von uns gewählten Problemen und Untersuchungsgegenständen umgehen, sie bedingen gleichzeitig auch die Möglichkeitsräume, wie Daten, Informationen und Erkenntnisse strukturiert, ausgewertet und repräsentiert werden. Die Ausführung von Prozessen ist keineswegs reibungslos, sondern stets eingebettet in Systeme, die sich diesen Ausführungen widersetzen oder ihnen neue Bedeutungen zuweisen. Eine reflexiv-kritische Hinwendung zu Software kann jedoch eine Abkürzung für Forschungsvorhaben darstellen, die an einem tieferen Verständnis der Phänomene rund um Code und Software interessiert sind – etwa um erziehungswissenschaftliche Erkenntnisse im Horizont der Digitalität zu gewinnen und eine disziplinäre Selbstvergewisserung vorzunehmen.

Referenzen

- Breiter, Andreas (2023). cmd. In Sandra Hofhues/Konstanze Schütze (Hg.), *Doing Research*. Bielefeld: Transcript, 163–167.
- Chun, Wendy H. K. (2011). *Programmed visions. Software and memory*. Cambridge et al.: MIT Press.
- Dander, Valentin (2014). Von der ‚Macht der Daten‘ zur ‚Gemachtheit von Daten‘. Praktische Datenkritik als Gegenstand der Medienpädagogik. *Mediale Kontrolle unter Beobachtung. Datenkritik*, 3(1), 1–21. <https://doi.org/10.25969/mediarep/13783>
- D'Ignazio, Catherine/Klein, Lauren F. (2020). *Data feminism*. Cambridge et al.: MIT Press.
- Foucault, Michel (1981). *Archäologie des Wissens*. Frankfurt/Main: Suhrkamp.
- Foucault, Michel (2019). *Die Ordnung des Diskurses*. 15. Aufl. Frankfurt/Main: Fischer Taschenbuch.
- Fuller, Matthe (Hg.) (2008). *Software studies. A lexicon*. Cambridge et al.: MIT Press.
- Galloway, Alexander R. (2004). *Protocol: How control exists after decentralization*. Cambridge et al.: MIT Press.
- Galloway, Alexander R. (2006). Language Wants To Be Overlooked. On Software and Ideology. *Journal of Visual Culture*, 5(3), 315–31.
- Galloway, Alexander R./Thacker, Eugene (2007). *The exploit. A theory of networks*. Bd. 21. Minneapolis et al.: ebrary.
- Golumbia, David (2009). *The cultural logic of computation*. Cambridge et al.: Harvard University Press.
- Jörissen, Benjamin/Verständig, Dan (2017). Code, Software und Subjekt. Zur Relevanz der Critical Software Studies für ein nicht-reduktionistisches Verständnis „digitaler Bildung“. In Ralf Biermann/Dan Verständig (Hg.), *Das umkämpfte Netz. Macht- und medienbildungstheoretische Analysen zum Digitalen*. Wiesbaden: Springer Fachmedien Wiesbaden, 37–50.
- Kittler, Friedrich A. (1992). There is no software. *Stanford Literature Review*, 9(1), 81–90.
- Lessig, Lawrence (1999). *Code and other laws of cyberspace*. New York: Basic Books.
- Lessig, Lawrence (2010). *Code. Version 2.0*. 2. Aufl. [S.l.]: SoHo Books.
- Mackenzie, Adrian (2005). The Performativity of Code. Software and Cultures of Circulation. *Theory, Culture & Society*, 22(1), 71–92.
- Manovich, Lev (2013). *Software Takes Command*. London: Bloomsbury Publishing.
- Marino, Mark C. (2020). *Critical Code Studies. Initial Methods*. Cambridge et al.: MIT Press.
- McIlwain, Charlton D. (2020). *Black Software. The Internet and Racial Justice, from the AfroNet to Black Lives Matter*. New York: Oxford University Press.
- Mittelstraß, Jürgen (2005). Methodische Transdisziplinarität. *Zeitschrift für Technikfolgenabschätzung in Theorie und Praxis*, 14(2), 18–23. doi: 10.14512/tatup.14.2.18
- Montfort, Nick/Bogost, Ian (2009). *Racing the Beam. The Atari Video Computer System*. Cambridge et al.: MIT Press.
- Montfort, Nick/Baudoin, Patsy/Bell, John et al. (2014). *10 PRINT CHR\$(205.5+RND(1)); : GOTO 10*. Cambridge et al.: MIT Press.
- Pritchard, Hele/Snodgrass, Eric/Tyzlik-Carver, Magda (Hg.) (2018). *Executing Practices*. Bd. 6. London: Open Humanities Press.
- Rieder, Bernhard/Röhle, Theo (2012). Digital Methods. Five Challenges. In David M. Berry (Hg.), *Understanding Digital Humanities*. London: Palgrave Macmillan UK, 67–84.
- van Schewick, Barbara (2010). *Internet Architecture and Innovation*. Cambridge et al.: MIT Press.
- Schäffer, Burkhard (2012). Erziehungswissenschaftliche Medienforschung. Medienpraxiskulturen im Generationenvergleich. In Friedrich Ackermann/Thomas Ley/Claudia Machold/Mark Schrödter (Hg.), *Qualitatives Forschen in der Erziehungswissenschaft* (S. 135–156). Wiesbaden: Springer.
- Soon, Winnie/Cox, Geoff (2020). *Aesthetic Programming. A Handbook of Software Studies*. London: Open Humanities Press.
- Swertz, Christian (2020). Big Data als datenbasierte Programmierung: Eine medienpädagogische Analyse künstlicher neuronaler Netzwerke. *MedienPädagogik*, 17, 93–119.
- Verständig, Dan (2020). Die Ordnung der Daten. Zum Verhältnis von Big Data und Bildung. In Stefan Iske/Johannes Fromme/Dan Verständig/Katrin Wilde (Hg.), *Big Data, Datifizierung und digitale Artefakte*. Bd. 42. Wiesbaden: Springer, 115–139.
- Wachter-Boettcher, Sara (2017). *Technically Wrong: Sexist Apps, Biased Algorithms, and other Threats of Toxic Tech*. New York: W.W. Norton & Company.
- Wardrip-Fruin, Noah (2012). *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Cambridge et al.: MIT Press.

