

Reinhard Storz

Texte, Scripts und Codes: Ein Exkurs zur Grenzfigur des Programmierer-Künstlers

2002-07-24

<https://doi.org/10.25969/mediarep/17554>

Veröffentlichungsversion / published version
Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:

Storz, Reinhard: Texte, Scripts und Codes: Ein Exkurs zur Grenzfigur des Programmierer-Künstlers. In: *Dichtung Digital. Journal für Kunst und Kultur digitaler Medien*. Nr. 24, Jg. 4 (2002-07-24), Nr. 4, S. 1–7. DOI: <https://doi.org/10.25969/mediarep/17554>.

Nutzungsbedingungen:

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

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

Terms of use:

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

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

Texte, Scripts und Codes: Ein Exkurs zur Grenzfigur des Programmierer-Künstlers

Von Reinhard Storz

Nr. 24 – 24.07.2002

Abstract

In der Medienkunst zeigt sich eine neue, noch seltene Grenzfigur: der Programmierer-Künstler. Er ist der Meister der Relationen und Interaktionen, der ifthen-Befehle und der objektorientierten Effekte. Die Programmiererin entwirft nicht Figuren, sondern Aktionen, und weil ihr Medium die 'unsichtbare' Schrift ist, Wörter und Zahlen, ist sie Spezialistin für die datenarme Arbeit im Internet. Die Schrift der Programmiererin kennt nur einen Leser: den Computer. An ihn ist ihr Skript adressiert. Er liest und befolgt den Befehl, und was wir vom Text schliesslich zu sehen bekommen, ist seine Interpretation.

Lassen sich Begriffe wie "Lesen" und "Interpretieren" auf die Tätigkeit von Computern und ihrer Programme anwenden? Sind die nicht sture Rechenknechte, die nur prozessieren und exekutieren, was ihnen im Imperativ eingeflüstert wird? Oder, anders gefragt, sind Lektüre und Interpretation nicht kulturelle Akte, die so etwas wie Eigenwillen voraussetzen? Und wenn der Code das kreative Medium der neuen Kunst ist, wenn Maschinen Kunst schaffen können, wer bzw. wo ist dann der Künstler?¹ Noch immer suchen wir den Pinselstrich des Künstlers auf der Leinwand.

Programmiererinnen sind Autoren der besonderen Art, Logikerinnen am Rand der Mathematik. Visionärer Überschuss und enigmatische Formeln sind unerwünscht. Und doch sind sie die modernen Zaubererinnen, oder scheinen das zu sein, was bei der Zauberei genügt. Sie murmeln hinten Zaubersprüche ins Programm und vorn am Bildschirm sehen wir Levitationseffekte. Doch wie bei jeder Magie gibt es unerbittliche Gesetze. Wird in der Zauberformel auch nur eine Silbe, ein Komma vergessen, ist die Geschichte kaputt, der Effekt gleich Null.

Die Figur des Zauberers mag im Kunstkontext traditionelle Bedürfnisse bedienen, über die Kommastelle ironisiert sie sich selbst. Die Esoterik der neuen Medien fusst

sowohl auf klassischen Technikmythen wie auf der traditionellen Auffassung der Kunstgeschichte, die in den Künstlerinnen weiterhin das inspirierte Medium des Numinosen sehen will. Doch der Programmierer-Künstler ist im Grund ein Ikonoklast, sein Metier ist die Schrift, nicht das Bild - die Handlung, nicht die Darstellung.

Noch immer verursachen Programmierer-Künstlerinnen und ihr Tun bei den Experten deshalb kunstbegriffliche Kopfschmerzen. In Kunstjürs werden sie oft als Maschinisten der Bildautoren bewertet, und auch bei Werken, in denen die User im Handlungsgerüst der Programmierung eigene Figuren oder Hypertextelemente beisteuern können, wird der Kunstwert eher produkt- als aktionsbezogen definiert.

Die Differenz zwischen Sprechen und Tun gehört zu den Schlüsselfragen der Kunsttheorie, und an dieser Bruchstelle stösst die neue Text- und Kunstform den Kunstexperten auf. In einem Interview mit Tilman Baumgärtel sagt der Medienkünstler John F. Simon jr: "Es liegt in der Natur der Software, daß man einen Quellcode schreibt, und der wird dann von einem Computer ausgeführt. Man könnte die Ideen einiger Konzeptkünstler tatsächlich als Programm schreiben, und dann von einem Computer ausführen lassen. Die Kunstwerke würden dann einfach sich selbst ausführen, oder, einfacher gesagt: die Kunst tut, was sie sagt."² Wenn Programme tun, was sie sagen - sagen sie auch, was sie tun? Und ist die Rede des Programmcodes in gleichem Mass vom Recht auf freie Meinungsäusserung geschützt, wie etwa die Rede in der Kunst?

Im New Yorker Gerichtsfall 'University City Studios, Inc. v. Eric Corley' werden diese Fragen in juristischen Überlegungen kontrovers diskutiert. Dabei zeigt sich im Kontext der juristischen Wahrheitsfindung, wie eng die Problematik der Programmierscripts mit unserem Kunstbegriff verbunden ist. Corpus Delicti im Gerichtsfall ist das sogenannte DeCSS-Script, mit dem sich der Kopierschutz für DVD-Filme knacken lässt. Das Script war von den angeklagten jungen Hackern über Websites veröffentlicht worden, und mit dem Verlust des Kopierschutzes drohten der amerikanischen Filmindustrie hohe Einnahmeverluste.

Die Verteidigung argumentierte klar für die Unterstellung aller Programmierertexte unter den Schutz des First Amendment,³ denn Quellcode sei heute ein effektives Mittel der menschlichen Kommunikation. Sie bezog sich auf eine frühere Festschreibung des Supreme Court: "'All ideas having even the slightest redeeming social importance', including those 'concerning the advancement of truth, science, morality, and arts' have the full protection of the First Amendment." Und weiter: "DeCSS is no less deserving of such protection than music, for which the language of expression is musical notes that are unintelligible to the average person, or any form of art, where the range of expression is virtually limitless. (See Hurly v. Irish-American Gay, Lesbian and Bisexual Group, 515 U.S. 557, 569, 1995, noting that the versatile scope of the First Amendment "unquestionably shield[s]" such diverse

forms of expression as the artwork of Jackson Pollack (sic), the music of Arnold Schoenberg, and the Jabberwocky verse of Lewis Carroll)."4

Ideen vermitteln sich über die "Rede", sei es die der Verbalsprache, der Malerei oder der Musik. Die Verteidigung zitierte einen Urteilspruch, in dem Rede definiert wird:

"[S]peech in any language consists of the 'expressive conduct' of vibrating one's vocal chords, moving one's mouth and thereby making sounds, or of putting pen to paper, or hand to keyboard. Yet the fact that such 'conduct' is shaped by language—that is, a sophisticated and complex system of understood meanings—is what makes it speech. Language is by definition speech, and the regulation of any language is the regulation of speech." (4)

Diesen Ausführungen widersprachen die Klägeranwälte: Der amerikanische Kongress selbst habe Entschlüsselungstechniken als modernes Äquivalent zu "Schlüsseln" bezeichnet. Das unerlaubte Benutzen eines gestohlenen Autoschlüssels könne aber nicht im Ernst vom Recht auf freie Meinungsäußerung geschützt sein, nur weil er Informationen mit dem Schlosszylinder austausche. Es müsse unterschieden werden zwischen der Rede, etwa der literarischen Beschreibung eines Verbrechens, und der Tat selbst, etwa der Anwendung eines Instruments im Tatvollzug.⁵ Die Verteidigung zitierte darauf einen früheren Prozess, in dem klargelegt worden war, dass Lochstreifen für ein mechanisches Klavier als Rede im Sinn des First Amendment zu gelten habe, weil der Musiker durch sie kommuniziere und auf das Instrument einwirke. Dass die "Rede" der Lochstreifenprogrammierung "funktional" sei, spiele dabei keine Rolle. Und weiter, als spöttische Replik der Verteidigung an die Kläger: "Defendants appear to insist that the higher the utility value of speech the less like speech it is. An extension of that argument assumes that once language allows one to actually do something, like play music or make lasagne, the language is no longer speech."⁴

Zur Funktionalität geschriebener Rede lassen sich weitere Gedanken anfügen. Auch Partituren und literarische Schriftstücke sind zuerst blosse Handlungsanweisungen, sie steuern neurologische und motorische Vorgänge. William M. Ivins beschrieb das (Vor-) Lesen von Texten in unserer Alphabet-Schrift 1953 als 'biomaschinellen' Vorgang: "Jedes geschriebene oder gedruckte Wort besteht aus einer Zeile konventioneller Anweisungen, gemäss denen man in einer spezifischen linearen Anordnung Muskelbewegungen ausführt, die bei richtiger Ausführung eine Abfolge von Lauten ergeben."⁶ Und Walter Ong schrieb 1982 über Partituren und den Akt ihrer musikalischen Interpretation: "Das moderne Orchester zum Beispiel ist ein Ergebnis hoher Technologie. Die Violine ist ein Instrument, gewissermassen ein Werkzeug. Eine Orgel ist eine riesige Maschine, mit Kraftquellen - Pumpen, Blasebälgen, Generatoren - alles dem Spieler vollkommen äusserliche Dinge. Beethovens Partitur zu seiner 5. Sinfonie besteht aus sorgfältigen Anweisungen für hochspezialisierte Techniker, in denen er genau vorgibt, wie die Instrumente zu

nutzen sind. Legato: Nimm die Finger nicht von der Taste, ehe du die nächste angeschlagen hast. Staccato: Schlag die Taste an und nimm die Finger sofort weg.¹⁷

Diese Verweise sprechen vom primären Lesevorgang an der Schnittstelle zwischen Schriftmedium und Mensch, vom Entziffern der Botschaft, die in der Aufreihung konventioneller Zeichen besteht: Noten, Buchstaben, Zahlen. Unsere westlichen Schriftsysteme sind erfolgreich, weil "sie selber nicht denken"¹⁸, weil zum Beispiel das Alphabet das rein passive Instrument des gesprochenen Wortes ist und keinen Unterschied macht zwischen Sens und Nonsens. Doch Sinn hilft. Und Maschinen stellen sich bei Unsinn taub. Die Notation in vielen Anwendungsprogrammen am Computer bedient sich neben Zahlen und Interpunktionszeichen der Alphabetsprache. Das ist eine Konzession der Programmentwickler an die Anwender, welche ihre Ideen in der programmspezifischen Scriptsprache realisieren. Die Software simuliert das Menschliche nach beiden Seiten hin, zum Scriptprogrammierer wie zum Werkbetrachter.

Im DeCSS-Fall befand das Gericht in erster Instanz, gestützt auf den 'Digital Millennium Copyright Act' sei die Verbreitung des Scripts zu verbieten, da die Funktionalität des Codes gegenüber seinem Redecharakter ausschlaggebend sei. Die Funktion des DeCSS-Scripts bestehe unzweifelhaft und primär darin, Zugang zu einem vom DMCA geschützten Werk zu schaffen.⁹ Bezogen auf die Veröffentlichung im Internet bemühte Richter Kaplan in der Begründung die Hilfe der Viren-Metapher: Wie bei Computer-Viren bestehe die Gefahr der epidemischen Verbreitung des inkriminierten Scripts im Netz. Vorsichtig fügte die Urteilsschrift aber an, es seien leicht Umstände denkbar, in denen der juristische Versuch, die Verbreitung von Computercode zu "regulieren", nicht in selbem Mass berechtigt wäre.¹⁰ Das Urteil vom 17. August 2000 wurde im Januar 2001 an den "U.S. Court of Appeals" in New York City weitergezogen.¹¹

Aktivisten der internationalen Opensource- und Copyleft-Bewegung reagierten prompt auf die Formulierung des New Yorker Urteils. Die Funktionalität des Codes wurde mit verstärkter Rede ummantelt. Wenn direkt anwendbarer Quellcode illegal ist, musste nicht-direkt anwendbarer demnach legal sein. So begannen die Aktivistinnen, den inkriminierten Code in dramatische Lesungen, Bilder, Filme, Musik und mathematische Formeln umzugießen.¹² Ob dabei gute Kunst entstand, ist zweitrangig, hauptsache Kunst; "DeCSS Kunst"¹³. Kunst dient hier als Mittel der Steganographie, als Verschlüsselung. Nach Jean-François Lyotards Definition von 1986 ist das ihr ureigenes Feld: "Und deshalb besteht die Arbeit des Künstlers gerade darin, Operatoren zu suchen, die sich zur Erzeugung von Sätzen eignen, die ihrerseits noch nie gehört worden sind und also definitionsgemäss - zumindest vorläufig - auch nicht mitteilbar sind. Mitteilbar werden diese Sätze erst, wenn die Operatoren, mit denen sie erzeugt werden können, auch ihren Empfängern bekannt sein werden und wenn diese sie rückübertragen können."¹⁴ Die Künstler erzeugen

also fremdartige Werke, deren Funktion ausschliesslich im Experimentieren mit den Regeln besteht.

Auch wenn Anlass und Form der DeCSS-Werke den Ausführungen Lyotards eine leicht komische Wendung geben, sie befolgen die sprachphilosophische Analyse wie eine Rezeptur. Ihr Clou besteht darin, das DeCSS-Script durch poetische oder mathematische Filter zu schicken und den Empfängern die Rückübertragung gerade so schwer zu machen, dass sie juristisch nicht verfolgt werden kann. Dass diese Werke dabei auch die "Gesetze der guten Kunst" verletzen (mit den Operatoren winkt man nicht wie mit einem Zaunpfahl!), schmerzt allenfalls deren Richter.

Die Wahrheit bleibt: In ihrer reinen Form sind Programmierertexte wie das DeCSS-Script apoetisch, auch wenn sie Kunst erzeugen. Ihre Autorinnen sind schroffe Schriftsteller. Sie kennen nichts als den reinen Befehlston: Tu dies und tu das, und wenn das nicht der Fall ist tu jenes. Dazu kommt ein Berufsethos der Wortkargheit, je knapper desto besser. Wenn einer das selbe in zwanzig Zeilen sagt, und der andere sagt es in zehn, ist der schnellere der bessere. Stilfragen sind nur für Eingeweihte erkennbar: die Reduktion des epischen Stoffes auf eine schlanke Zeitachse, die Ästhetik der blanken Stringenz.¹⁵

Viele Kuratorinnen und Kunstkritiker tun sich heute noch schwer damit, die Arbeit der Programmierer-Künstlerinnen als kunstrelevant zu beachten. Sie schauen nicht hin und hoffen, damit die Grenzfines der Kunst und das Interesse ihrer Institutionen zu schützen. Mit den Werken der neuen Medienkunst ignorieren sie auch deren Distributionskanäle, sie fürchten die Folgen der neuen Kommunikationstechnologien und - einmal mehr - die drohende Auflösung des Autorenbegriffes in der Kunst. Damit haben sich die Programmierer-Künstlerinnen auseinanderzusetzen.

Der russische Medienkünstler Vladimir Muzhesky betonte in einem Gespräch, die alte Rivalität zwischen Kunst und Technologie, die Idee der Kunst als Technologiekritik sei ihm fremd. "Ich glaube einfach nicht mehr, dass es ein Monopol der Kreativität oder Produktivität gibt. Die zeitgenössische Situation ist die der Dispersion, der Zerstreuung."¹⁶ Aus der Perspektive eines Kurators am Guggenheim-Museum kommentiert Jon Ippolito: "Was wir heute miterleben ist weder eine Implosion noch eine Zersplitterung der Kunst, sondern vielmehr eine seismische Instabilität entlang der Bruchlinie zwischen Kunst und Nicht-Kunst. Um diese Volatilität zu bewältigen, muss unsere Definition von Kunst auch für Beispiele kunstähnlicher Kreativität offen sein, die in Disziplinen ohne direkten Bezug zur Kunstgeschichte oder zum Kunstmarkt entstanden sind." Und weiter: "Jüngere Generationen von KunstkritikerInnen, KuratorInnen und PhilosophInnen können nicht mehr durch blosses Hinsehen beurteilen, ob etwas Kunst ist oder nicht."¹⁷

Die Programmiererbande pflegt den oppositionellen Touch im Betriebssystem Kunst wohl nicht ungerne. Mit Begriffen wie 'Proceduralism' (Judson Rosebush,

bereits 1989) oder 'Generative Art' (Cornelia Sollfrank) schafft sie aber Etiketten, die in den bestehenden Kunstbetrieb leicht integriert werden könnten, denn der ist angewiesen auf neues theoretisches 'Branding'. Nur: die neuen digitalen Kunstprodukte haben es, was ihre Distribution betrifft, nicht mehr nötig, in den traditionellen Kunstbetrieb eingemeindet zu werden. Sie verfügen über ihren eigenen Publikationskanal, das Netz, und sie haben bereits ihr eigenes, wachsendes Publikum.

Anmerkungen

1. Gemeint ist hier nicht nur das Programmieren in Programmier-Hochsprachen, sondern auch in Scriptsprachen wie dhtml, Java Script, Lingo oder Action Script (Flash 5). Der Streit, ob etwa "Java Script" als Programmiersprache im eigentlichen Sinn definiert werden kann, muss hier nicht weiter interessieren. (vgl. dazu die Diskussion unter: http://www.teamone.de/selfhtml/sfar-chiv/1999_2/t03805.htm#a18421).
2. Tilman Baumgärtel: Die New York School der Programmierer (<http://www.heise.de/tp/deutsch/inhalt/sa/3341/1.html>). Von John Simmons stammt die Online-Arbeit 'Every Icon', welche systematisch jedes mögliche aus 32 mal 32 Bildpunkten zusammengesetzte Bild generiert.
3. Erster Zusatzartikel zur Amerikanischen Verfassung, der die freie Rede ("Free Speech") garantiert.
4. Alle direkten und indirekten Zitate der Verteidigung unter: http://www.eff.org/IP/Video/MPAA_DVD_cases/20000808_ny_post_trial_brief.html
5. Alle direkten und indirekten Zitate der Anklage unter: <http://www.mpa.org/Press/DeCSS2.htm>
6. William Ivins: Prints and Visual Communication, Cambridge, MIT Press 1953, S.54-55. Übersetzung in: Der McLuhan-Reader, Mannheim 1997, S. 85
7. Walter Ong: Oralität und Literalität, Die Technologisierung des Wortes. Auszug, abgedruckt in: Kursbuch Medienkultur, Die massgeblichen Theorien von Brecht bis Baudrillard, Stuttgart 1999, S.99
8. Eric A. Havelock: Gesprochener Laut und geschriebenes Zeichen. Auszug, abgedruckt in: Kursbuch Medienkultur, Die massgeblichen Theorien von Brecht bis Baudrillard, Stuttgart 1999, S.91

9. Urteilstext: <http://www.nysd.uscourts.gov/courtweb/pdf/D02NYSC/00-08592.PDF> (S.28-34)
10. Urteilstext: <http://www.nysd.uscourts.gov/courtweb/pdf/D02NYSC/00-08592.PDF> (S.63-68)
11. Programmatischer Text der Community für die Eingabe ans Appellations-Gericht: <http://www-2.cs.cmu.edu/~dst/DeCSS/amicus-brief.html>
12. Unter <http://www.cs.cmu.edu/~dst/DeCSS/Gallery/> findet man alle wichtigen Links zum Prozess und zu den künstlerischen Folgeprojekten. Zu den Transformationen steht folgende Begründung: "If code that can be directly compiled and executed may be suppressed under the DMCA, as Judge Kaplan asserts in his preliminary ruling, but a textual description of the same algorithm may not be suppressed, then where exactly should the line be drawn? This web site was created to explore this issue, and point out the absurdity of Judge Kaplan's position that source code can be legally differentiated from other forms of written expression."
13. Jon Ippolito benutzte den Begriff "DeCSS art" in seinem Eingangsreferat zur Ars Electronica 2001. Vgl. Joline Blais / Jon Ippolito: Wie man Kunst immer am falschen Ort sucht. (Looking for art in all the wrong places) in: Takeover - wer macht die Kunst von morgen. Ars Electronica 2001. Wien, New York 2001, S.34
14. Jean-François Lyotard: 'Regeln und Paradoxa' in: Philosophie und Malerei im Zeitalter ihres Experimentierens, Berlin 1986; S.99-100
15. Der sprachstilistische Ehrgeiz der Programmiererinnen erinnert eher an Sport als an poetischen Sprachgebrauch. Allerdings hat der Weltrekordler Lynford Christie behauptet, auch seine 100 Meter-Sprints seien pure Posie.
16. Der russische Medienkünstler Vladimir Muzhesky im Gespräch mit Barbara Basting. www.xcult.org/shrink/text/basting_d.html
17. Jon Ippolito in seinem Eingangsreferat zur Ars Electronica 2001. Vgl. Joline Blais / Jon Ippolito: Wie man Kunst immer am falschen Ort sucht. (Looking for art in all the wrong places) in: Takeover - wer macht die Kunst von morgen. Ars Electronica 2001. Wien, New York 2001, S.40