Über Literatur und Digitalcode

Florian Cramer

Freie Universität Berlin, Institut für Allgemeine und Vergleichende Literaturwissenschaft
9/2001 (überarbeitet 2002 und 2003)

ABSTRACT

Diesem Text liegt die Hypothese zugrunde, daß die theoretische Diskussion digitaler Literatur, ebenso wie die Dichtung, auf die sie sich bezieht, ihren Focus verschoben hat von computerprozessierten Zeichen als vermeintlicher Erweiterung und Überschreitung des Textes (manifest in Begriffen wie „Hypertext“, „Hyperfiction“, „Hyper-/Multimedia“) hin zur Textlichkeit von Computerdaten an sich. Einige Phänomene, die dies praktisch belegen:

- Die Auseinandersetzung mit der Ästhetik und Politik von Computercodes in der unter dem Label „net.art“ bekanntgewordenen konzeptualistischen Netzkunst;
- im Gegenzug die Prägung experimenteller Netzdichtung durch net.art-Ästhetik;
- die institutionelle und personelle Affinität von net.art zum politischen Aktivismus im Netz;
- die enge ästhetische Anlehnung der net.art an die Sprachen und Codes einer älteren, rein technisch ausgerichteten „Hacker“-Kultur
- das Zusammenwachsen der drei genannten Kulturen - Netzkunst, Netzaktivismus und Hackerkultur;
- Freie Software bzw. Open Source und offene Netzwerkprotokolle als Schlüsselthemen der ästhetischen und politischen Diskurse aller dreier Kulturen;
- schließlich der Einfluß von Hacker-, net.art-, Code- und Netzprotokoll-Ästhetik auf zeitgenössisches Schreiben im Internet (z.B. bei mez, Alan Sondheim, Talan Memmott und Ted Warnell).

So fragt sich, wie „Codeworks“ - wie sie Alan Sondheim nennt - unter Textbegriffe faßbar sind, die ohne Reflexion von digitalem Code im allgemeinen und digitalem Programmcode im besonderen geprägt wurden. Ist es bloß ein Zufall, daß Codeworks, die Lowlevel-Codes von Computerprogrammen und Netzwerkprotokollen ästhetisch appropriieren, äußerlich konkreter Poesie annähern? In welchem Verhältnis stehen Computerprogramme und Literatur darüber hinaus? Ist Softwarekunst literarisch, weil sie aus Quelltexten entsteht?

17 November 2003

Über Literatur und Digitalcode

Florian Cramer

Freie Universität Berlin, Institut für Allgemeine und Vergleichende
Literaturwissenschaft
9/2001 (überarbeitet 2002 und 2003)

1. Code

Da Computer und Internet wie alle digitalen Technologien mit Nullen und Einsen operieren, basieren sie auf Code. Nullen und Einsen sind ein Alphabet, das informationsverlustfrei in andere Alphabete übersetzt und von ihnen ebenso informationsverlustfrei rückübersetzt werden kann. -- Es ist unsinnig, die Defi nition von ,,Alphabet‘‘ im allgemeinen auf das lateinische Alphabet im besonderen zu begrenzen, da dieselbe Textinformation arbiträr als Buchstaben, Morse- ode, Flaggenzeichen oder Sequenzen von Nullen und Einsen codiert werden kann. Während das Alphabet Computern und Internet technisch eingeschrieben ist, können z.B. Bilder und Töne erst dann digital gespeichert und prozessiert werden, wenn sie von analogen Licht- und Tonsignalen in numerischen Code übersetzt wurden, eine Übersetzung, die im Gegensatz zur digitalen Speicherung konventioneller Texte, verlustbehaftet, also irreversibel und asymmetrisch ist. Töne und Bilder sind per se nicht codiert, sondern müssen in Codes gerastert werden, um computerberechenbar zu sein; jeder Text hingegen, der mit einem Setzkasten oder einer Schreibmaschine geschrieben werden kann, ist bereits codiert. Literatur ist daher in digitalen Informationsprozessoren eine privilegierte symbolische Form. Man kann eine Sammlung von Textdateien nach dem Wort ,,Vogel‘‘ durchsuchen, während die Suche nach Vögeln in digitalisierten Bilder oder von Vogelstimmen in digitalisierten Tönen unvergleichlich aufwendiger und fehleranfälliger ist, da sie entweder künstliche Intelligenz-Algorithmen oder manueller Indizierung voraussetzt, beides Methoden, die unsemantischen Text (numerische Werte für Graphik-Pixel bzw. Tonwellen-Ausschläge) in semantischen (nämlich Wortbeschreibungen) übersetzen.

Ebenso gilt das Umgekehrte: Digitale Daten und Algorithemn können auch in nichtdigitalen Medien wie z.B. Büchern gespeichert werden, solange sie in Zeichen übersetzt werden, die nach der Logik eines Alphabets codiert sind, wie es z.B. in Programmierhandbüchern und technische Standardisierungsdokumenten des Internets geschieht. Für die verlustfreie Übersetzung von Computercode in Druckbuchstaben mit anschließender Rückübersetzung der Buchstaben in Computerdateien gibt es, bis heute, zwei prominente Beispiele:

1 den Quellcode von Phil Zimmermans Verschlüsselungssoftware ,,Pretty Good Privacy‘‘ (PGP).

Die PGP-Algorithmen galten unter U.S.-amerikanischen Recht als Kriegswaffen und unterlagen daher einem Exportverbot. Zimmerman umging es, indem er den Programm-Quelltext von PGP in einem Buch veröffentlichte. Da Bücher, anders als Algorithmen, unter den ersten amerikanischen Verfassungszusatz der Redefreiheit fallen, konnte der PGP-Code in andere Länder exportiert und, durch Scannen und Abtippen, in ein maschinenausführbares Programm rückübersetzt werden.

2 der Quellcode von DeCSS, einem kleinem Programm, das die Verschlüsselung von DVD-Filmen knackt.

Da DeCSS laut dem amerikanischen Digital Millennium Copyright Act (DMCA) und dem kürzlich novellierten europäischen Urheberrecht illegal ist, wurden in den USA auch Flugblätter und T-Shirts mit aufgedrucktem DeCSS-Quellcode verboten.

Die Ansicht, daß Programmcode ,,freie Rede‘‘ (im Sinne auch von Schrift) sei, ist unter Informatikern und Programmierern weit verbreitet und bildet auch die Grundlage von Lawrence Lessigs juristischer Theorie des Internets.[1] Strenggenommen ist es falsch, von ,,digitalen Medien‘‘ zu reden, da es nur digitale Information gibt, die in analogen Medien wie Elektrizität, Lichtwellen und Magnetfeldern übertragen und

[1] Lessig, Lawrence, *Code and Other Laws of Cyberspace*, Basic Books, New York (2000).

gespeichert wird. Digitale Information wird ,,medial‘‘ nur durch analoge Technik; Computerbildschirme, Lautsprecher, Drucker zum Beispiel sind analoge Ausgabegeräte, die an Computern durch Digital-Analog-Wandler wie Video- und Audiokarten gekoppelt sind.[1]

Ein heute typischer PC nutzt magnetische Scheiben (Disketten und Festplatten), optische Scheiben (CD-ROM und DVD-ROM) und Chipspeicher (RAM, USB-Flash-Speicher) als seine Speicher- und elektrische oder optische Leiter als seine Übertragungsmedien. Theoretisch könnte man aber auch einen Computer mit eingebautem Scanner und Drucker bauen, der Bücher und Druckschriften als Speichermedien verwendet. Alan Turings Maschine zeigt, daß keine Elektronik nötig ist, um einen Computer zu bauen; im Bostoner Computermuseum steht ein einfacher mechanischer Computer, der aus Besenstielen zusammengebaut wurde.

Gegenüberstellungen von ,,dem Buch‘‘ und ,,dem Computer‘‘ sind irreführend, weil sie die Speicher- und Ausgabemedien (Papier versus einer Vielzahl optischer, magnetischer und elektronischer Techniken) mit der Information (alphabetischer Text versus binärer Code) verwechseln. Sie ignorieren auch den Reichtum traditioneller Speicher- und Übertragungsmedien der Literatur, die über das Buch hinaus auch mentale Speicherung und mündliche Übertragung, Tonkonserven, Radio und Fernsehen einschließen, um nur einige zu nennen. Eben weil Literatur per ihrer Definition als ,,Buchstabenkunst‘‘ codiert ist, ist sie (anders als z.B. Malerei und Skulptur) nicht an ein bestimmtes Material gebunden und kann über faktisch jedes Medium verbreitet werden.

Wenn es im strengen Sinne keine digitalen Medien gibt, gibt es strenggenommen auch keine digitalen Bilder und Töne. Ein sogenanntes ,,digitales Bild‘‘ ist ein Sequenz von Codes, die Maschinenbefehle erzeugen, welche wiederum den Elektrizitätsfluß erzeugen mit dem einer analoger Bildschirm oder Drucker ein Bild ausgibt.[2]

Natürlich ist es nicht unerheblich, ob z.B. eine Kette von Nullen und Einsen in eine Bilddarstellung übersetzt wird, weil dies ihre Semantik und somit Interpretation und Anwendung bestimmt. Meine vorige Argumentation sollte dies, obgleich formalistisch, nicht verleugnen, sondern lediglich unterstreichen, daß

1 wenn von ,,Multimedia‘‘ oder ,,Intermedia‘‘ in Verbindung mit Computern die Rede ist, es nicht um digitale Systeme als solche geht, sondern um Übersetzungen digitaler Information in analoge Ausgaben und umgekehrt. Literatur wird ,,multimedial‘‘ also gerade nicht durch Digitalisierung, sondern durch Entdigitalisierung.

2 Text und Literatur in diesen Übersetzungsprozessen privilegierte symbolische Systeme sind, weil sie (a) schon codiert vorliegen und (b) Computer auf der Grundlage codierter Instruktion operieren.

Literatur und Computer treffen sich erstens dort, wo Alphabet und Code, Umgangssprachen und Programmiersprachen sich überschneiden, zweitens in der Ansteuerung analoger Technik durch digitalen Code. Während Code für Leser natürlich nicht existiert ohne die Medien, die ihn wahrnehmbar machen, erweitert umgekehrt der Computer die Medien der Literatur nicht. Sie alle - Elektrizität, elektrische Ton- und Bildübertragung etc. -- hat es auch vor und unabhängig von Computern und digitaler Informationsverarbeitung gegeben.

Um einen Standpunkt zu revidieren, den ich in einer älteren Publikation vertreten habe:[3] Das Konzept der digitalen Dichtung, oder Computernetzdichtung, impliziert keine spezifischen Medien, nicht einmal spezielle Maschinen. Wenn Computer aus Besenstielen gebaut und mit Schnürsenkeln oder Bongotrommeln vernetzt werden können (siehe <http://eagle.auc.ca/~dreid/>); wenn digitale Daten, einschließlich ausführbarer Algorithmen, in Büchern gedruckt und von ihnen zurück in Maschinen

[1] Tastaturen, Mäuse, Scanner und Kameras sind, reziprok, Analog-Digital-Wandler.

[2] Normalerweise zerfällt dieser Code in drei Teile. Der erste - die sogenannte Ton- oder Bilddatei - enthält die maschinen- und betriebssystemunabhängigen abstrakten Informationen, der zweite - die sogenannte Betrachter- oder Abspielsoftware - die Befehle, die die abstrakte Information in ein betriebssystemspezifisches Format umwandeln, der dritte - das Betriebssystem - wandelt die Programmausgabe für das Ausgabegerät, ob Bildschirm oder Drucker. Diese drei Programmschichten sind jedoch nur Konventionen. Theoretisch könnte die ‘‘digitale Bilddatei‘‘ selbst allen Befehlscode enthalten, um sich auf analogen Endgeräten auszugeben, einschließlich jenes Codes, der konventionell als Bootloader und Betriebssystemkern definiert ist.

[3] ‘‘Warum es zuwenig interessante Computerdichtung gibt. Neun Thesen’’, <http://userpage.fu-berlin.de/~cantsin/homepage/writings/net_literature/general/karlsruhe_2000//karlsruher_thesen.pdf>

eingespeist oder alternativ von Lesern mental ausgeführt werden können, kann Computernetzdichtung ebenso gut auch im Medium Buch erscheinen.

Der Terminus digitaler ,,Multimedialität“ oder ,,Intermedialität“ könnte hilfreicher neudefiniert werden als *die Möglichkeit Information von einem Medium ins andere verlustfrei hin und her zu übersetzen, so daß die sicht-, hör- oder fühlbare Repräsentation dieser Information arbiträr wird.* Dies ist nicht möglich, ohne daß die Information in irgendeiner Art von Alphabet - ob alphanumerisch, binär, hexadezimal oder in Morsezeichen - codiert ist.

2. Literatur

2.1. Synthese: Dinge zusammensetzen

Betrachtet man die textuelle Codiertheit digitaler Information, so läuft man natürlich Gefahr, an digitalem Code gemachte Beobachtungen auf Literatur als solche zu projizieren. Computer operieren mit einer Sprache, die keine Semantik kennt und die auch syntaktisch weitaus weniger komplex ist als menschliche Umgangssprache. Da die Alphabete von Computer- und Umgangssprachen wechselseitig konvertierbar sind, bleibt ,,Text“ - definiert als eine zählbare, endliche Menge alphabetischer Zeichen - eine gültiger Begriff für Sequenzen von Maschinencode und konventionelle Schrift gleichermaßen. In Syntax und Semantik jedoch sind beide nicht austauschbar. Computeralgorithmen sind, als logische Aussagen, eine formale Sprache und somit nur eine begrenzte Untermenge von Sprache insgesamt.

Es ist jedoch ein populärer Irrtum zu glauben, daß ,,Maschinensprachen“ nur für Maschinen lesbar seien und deshalb irrelevant für menschlich geschaffene Kunst und Literatur und daß auch umgekehrt Literatur und Kunst mit formalen Sprachen nichts zu tun hätten. Schließlich sind Computercodes und -programme keine Maschinenkreationen und Maschinenprozesse, die nur miteinander kommunizieren; sondern sie sind von Menschen geschaffene Schriften.[4] Der Künstler und Programmierer Adrian Ward plädiert für eine Umkehrung der Annahme, daß die Maschine eine Sprache beherrsche:

> ,,Wir sollten meiner Meinung nach eher in dem Sinne denken, daß wir unsere schöpferische Subjektivität in automatisierte Systeme einschreiben, anstatt naiv zu versuchen, einem Automaten seine ‚eigene‘ kreative Agenda zu verschaffen. Viele von uns tun dies tagein und tagaus. Wir nennen es ‚Programmieren‘“.[5]

Man könnte dies auch Komposition von Partituren nennen, und es ist kein Zufall, daß Tonkünstler Computer viel früher eingesetzt und gründlicher verstanden haben als Dichter. Die westliche Musik ist das herausragende Beispiel einer Kunst, die auf einem schriftlich notierten, formalen Instruktionscode beruht. Ironien im Medium des Code wie ,,B-A-C-H“ bei Johann Sebastian Bach, die notentypographischen Figuration von Erik Saties ,,Sports et divertissements“ und die experimentellen Partituren von John Cage zeigen, daß formaler Steuercode eine eigene ästhetische Dimension und intellektuelle Komplexität besitzt, und nicht bloß Mittel zum Zweck eines Kunstwerks ist, das in einem anderen Medium erscheint. In einigen Werken sind Musikkomponisten von der klassischen Notenschrift zur natürlichen Schriftsprache gewechselt, so z.B. auch La Monte Young in seiner ,,Composition No.1 1961“, die sich einfach in der Anweisung ,,Draw a straight line and follow it.“ erschöpft.[62] Den Fluxus-Performances von George Brecht, George Maciunas, Nam June Paik und anderen liegen ähnliche Partituren zugrunde. 1969 schrieb der amerikanische Komponist Alvin Lucier sein zum Klassiker der neuen Musik gewordenes Stück ,,I am sitting in a room“ als eine kurze, gesprochene Anweisung, das Stück aufzuführen, in dem man sie selbst im

---

[4] Kein Computer kann sich selbst reprogrammieren. Selbstprogrammierung ist nur möglich innerhalb eines Rahmens von Spielregeln, die ein menschlicher Programmierer vorgegeben hat. Eine Maschine kann sich anders als erwartet verhalten, weil beim Verfassen ihrer Spielregeln nicht alle möglichen Situationen bzw. Systemzustände übersehen wurden, doch kann keine Maschine ihre Spielregeln überschreiben.

[5] ,,I would rather suggest we should be thinking about embedding our own creative subjectivity into automated systems, rather than naively trying to get a robot to have its ‘own’ creative agenda. A lot of us do this day in, day out. We call it programming.“, zitiert aus einer E-Mail an die Netzkunst-Mailingliste ,,Rhizome“ vom 7.5.2001

[2] Galerie und Edition Hundertmark, *George Maciunas und Fluxus-Editionen,* Galerie und Edition Hundertmark, Köln (1990).

Raum abspielt, wieder neu aufnimmt und dadurch akustisch moduliert.

In der Literatur sind formale Instruktionen notwendige, implizite Voraussetzung aller permutationellen und kombinatorischen Dichtung,[7]
deren Geschichte ihrerseits auf die Tradition der Buchstabenkombinatorik in der Kabbala und Sprache der Magie zurückgeht. Selbst in einer realistischen Erzählung gibt es eine implizite formale Anweisung wie - z.B. in welcher Reihenfolge - ein Text zu lesen sein, im Gegensatz zum Hypertext, der zwar alternative Reihenfolgen anbietet, zugleich aber seine Alternativen dem Leser (der in einem Buch noch frei blättern kann) zwingend vorschreibt. Und schließlich ist die Grammatik eine implizite, pervasive formale Instruktion.

Die Gramatik des Computers unterscheidet sich jedoch von der Grammatik natürlicher Sprache darin, daß, um einen informatischen Fachausdruck zu benutzen, der Namensraum von Instruktion und ihrer Ausführung derselbe ist: Wenn, wie Inke Arns es vorschlägt, man vom Instruktionscode als ,,Genotext'' und vom Code, den er erzeugt, als ,,Phänotext'' spricht, dann zeichnet sich computerprozessierte gegenüber gesprochener Sprache dadurch aus, daß sowohl Geno-, als auch Phänotext im selben Alphabet von Bits und Bytes codiert sind und im selben Medium von (heute in der Regel) elektrischen Strömen, während in gesprochener Sprache der Genotext der Grammatik im Geschriebenen implizit und nicht explizit steckt. Einem zufälligen Schnipsel digitalem Code kann nicht abgelesen werden, ob es maschinenausführbar ist oder nicht, ob es ein Genotext ist oder ein Phänotext. Jeder digitale Code, sogar ein ,,Projekt Gutenberg''-Text von, zum Beispiel, Homers ,,Odyssee'' ist potentiell maschinenausführbar, je nach dem, ob anderer Code - ein Compiler, Runtime-Interpreter oder die Logikschaltungen eines Mikroprozessors -- ihn als Kette von Maschinenbefehlen auszuführen, zumal Computercode in hohem Maße rekursiv ist und, als Architektur virtuell implementierter Maschinen, dazu tendiert, in Schichten um Schichten seiner selbst angehäuft zu werden.

2.2. Analyse: Dinge auseinandernehmen

Daß man digitalem Code nicht ansieht, ob es ein maschinenausführbares Programm ist oder nicht, ist conditio sine qua non aller E-Mail-Viren, aber auch der ,,Codeworks'' von Künstlern wie Jodi, antiorp/Netochka Nezvanova, mez, Ted Warnell, Alan Sondheim, Kenji Siratori - um nur einige Begründer des Genres zu nennen. Nur sind die Codeworks fiktional insoweit sie nur ästhetisch vorgeben, viraler Maschinencode zu sein.

Codeworks - ein Begriff, der von Alan Sondheim geprägt wurde - sind die herausragenden Beispiele einer digitalen Dichtung, die die interne Textualität des Computers reflektiert. Sie tun dies jedoch nicht, indem sie, um Alan Turing via Raymond Queneau zu zitieren, Computerdichtung für Computer sind,[83] sondern indem sie mit den Verwirrungen und Grenzunschärfen von maschinen- und menschengenerierter Sprache spielen und die kulturellen Implikationen dieser Überschneidungen reflektieren. Die ,,mezangelle''-Dichtung von mez (Mary-Anne Breeze), die die Syntax von Programmiersprachen, Netzwerkprotokollen und Englisch zu einer hybriden Phantasiesprache verschmilzt, ist ein herausragendes Beispiel solch einer Poetik.

Von älterer Formalcode-Dichtungen wie La Monte Youngs *Composition 1961* , Fluxus-Partituren und Anagramm- und Wortpermutationslyrik weichen Codeworks in einer bemerkenswerten Hinsicht ab: Sie konstruieren oder synthetisieren ihre Instruktionen nicht wie im Labor, sondern benutzen benutzen existierenden Code bzw. existierende Grammatiken, um sie zu zerlegen. Ich stimme Friedrich Block und seinen ,,Eight Digits of Digital Poetry'' <http://www.dichtung-digital.de/2001/10/17-Block/index-engl.htm> zu, daß digitale Dichtung im historischen Kontext experimenteller Lyrik betrachtet werden sollte. Eine Poetik der Synthese kennzeichnete kombinatorische Dichtung und künstlerische Instruktions-Partituren, eine Poetik der Analyse, des Auseinandernehmens, kennzeichnete den Dadaismus und seine Nachfolger. Jedoch findet man in den Vor- und Nachkriegs-Avantgarden des 20. Jahrhunderst nur wenig Dichtung, die mit formalen Instruktionen analytisch verfährt (wie z.B. die ALGOL-Computersprachenlyrik

[6] Einige historische Beispiele wurden auf meiner Website <http://userpage.fu-berlin.de/~ cantsin/permutations> als Computerprogramme adaptiert, die online ausgeführt werden können.

[7] Mit dem Virus ''biennale.py'' des Netzkünstlerduos <http://www.0100101110111001.org> als der Ausnahme eines Code-Kunstwerks, das tatsächlich ein Computervirus ist.

[3] Queneau, Raymond, *Cent mille milliards de poèmes*, p. 3, Gallimard, Paris (1961).

der Oulipo-Dichter François le Lionnais and Noël Arnaud in den frühen 1970er Jahren).[4] Die Codelyrik des Internets hingegen ensteht unter der historisch neuen -- wenn man so will, postmodernen - Bedingung einer flutenden Abundanz von Maschinencodes.

Die These, daß es keine digitalen Medien, sondern nur in analogen Medien gespeicherten und übertragenen digitalen Code gibt, findet sich in den Codeworks voll bestätigt. Das bevorzugte Medium der meisten genannten Code-Künstler ist, ganz im Gegensatz zu Hypertext und Multimedia, simpler ASCII-Text. Dabei sind komplexe technopoetische Reflexionen und Low Tech-Medien eben kein Widerspruch. Im Gegenteil ist Low Tech Teil des kritischen Selbstverständnisses der Codework-Künstler.

Hypertext- und Multimediadichter, deren Arbeiten historisch kurz vor dem World Wide Web und dann parallel zu seinem Ausbau entstanden, betrachteten sich zu Recht als dessen Pioniere. In den 1990er Jahren loteten sie fortwährend die technischen Grenzen sowohl des Internets, als auch von Multimedia-Computertechnik aus. Doch indem digitale Kunst und Literatur zu Technologiestudien neuer Browser-Funktionen und Multimedia-Plugins wurde, begab sie sich zugleich in die Sackgasse geschlossener, industriekontrollierter Codierungen.[9]
Ob beabsichtigt oder nicht, wurde digitale Kunst deshalb zur Komplizin der Umformatierung des World Wide Web von einem offenen, Betriebssystem- und Browser-agnostischen Informationsnetzwerk zu einer von proprietärer Technik abhängigen Plattform.

Da Codeworks den Leser umlenken von Softwarecode, der vorgibt, keiner zu sein (wie z.B. 3D-Simulationen), auf Code, der sich auch als Code präsentiert, haben sie klare Affinität zu Hacker-Kulturen, und zwar in technischer, ästhetischer und politischer Hinsicht. Zwar bezeichnet das Wort ,,Hacker‘‘ weitaus gegensätzlichere Akteure, als es in seiner Einheit suggeriert,[10]
doch könnten sie ebenfalls unterschieden werden in solche, die Dinge konstruktiv zusammenfügen, wie z.B. Freie Software- and Demo-Programmierer [5] und denjenigen, die Dinge zerlegen, wie z.B. Cracker von Seriennummern und Telekommunikationsnetzen aus dem Umfeld der Underground-Publikationen YIPL, TAP, Phrack und 2600. De facto stammen die wichtigsten poetischen Formen der Code-Künstler aus Hacker-Subkulturen seit den 1970er Jahren, so z.B. auch ASCII Art, Code-Slang[11] und Dichtung in Programmiersprachen (wie Perl poetry). Manche Codekünstler (wie z.B. jaromil und Walter van der Cruijsen) gehören sogar sowohl zur ,,hacker‘‘-, als auch zur ,,Netzkunst‘‘-Kultur.

Seit ihren Anfängen war die unter dem Label ,,net.art‘‘ bekanntgewordenen konzeptualistische Netzkunst Teil einer kritisch politisierten Netzkultur und ist auch heute noch eng verknüpft mit einem netzkulturellen Diskurs, wie ihn die internationale Mailingliste ,,Nettime‘‘ geprägt hat. In ihrer Ästhetik, Poetik und Politik wurzeln die Codeworks eher in der net.art als in der Hypertext-Literatur und ihren historischen Wurzeln an der amerikanischen Brown University.

Welche Verbindungen gibt es noch zwischen Literatur und digitalem Code, wenn man von den Codework-Poetiken absieht? Diskutiert man die Poetik des digitalen Codes als literarische Poetik - und nicht Literatur als digitalen Text --, so erweisen sich beide als eng miteinander verwandt. Dies bedeutet nicht, wie John Cayley in seinem Abstract zur Erfurter ,,p0es1s‘‘-Konferenz meint,[12] , daß man Friedrich Kittlers technozentrische Medientheorie unterschreibt, die, trotz ihrer Provokation der Geisteswissenschaften und des philosophischen Idealismus, in jene metaphysische Falle geraten zu scheint, die Jacques Derrida in ,,Écriture et différence‘‘ beschreibt: Durch die Ersetzung eines metaphysischen Zentrums (bei Kittler: der Geist, der aus den Geisteswissenschaft ausgetrieben wird) durch ein anderes (Technik und ihre Diskursgeschichte) wird Metaphysik nicht, wie behauptet, zerstört, sondern nur unter anderem Vorzeichen fortgeschrieben.

Im Untertitel stellt dieser Text eine offene Frage: ,,Können Textbegriffe, die ohne Berücksichtigung elektronischer Texte entwickelt wurden, auf digitalen Code angewandt werden, und welche Rolle spielt

---

[4] *Oulipo Compendium*, p. 47, Atlas Press, London (1998).

[8] wie z.B. die Datenformate und Softwaretechnologien Shockwave, QuickTime und Flash

[9] Boris Gröndahls Artikel ,,The Script Kiddies Are Not Alright‘‘ beschreibt die vielen, z.T. sogar antagonistischen Lager und Generationen des ,,Hackertums‘‘, <http://www.heise.de/tp/deutsch/html/result.xhtml?url=/tp/deutsch/inhalt/te/9266/1.html>

[5] Levy, Steven, *Hackers*, Project Gutenberg, Champaign, IL (1986 (1984)).

[10] wie ,,7331 wAr3z d00d‘‘ für ,,leet [=elite] wares dood‘‘

[11] http://www.p0es1s.net/poetics/symposion2001/a_cayley.html

Literatur dabei?‘‘ Ich möchte sie nur vorläufig beantworten: Während alle Literatur, die schriftlich codiert ist, die sprach- und textliche Verfaßtheit des Computers und digitaler Dichtung zu verstehen hilft, kann mit Computer und digitaler Dichtung als Denkfiguren die Codiertheit der Literatur und ihrer sprachlichen Kontrollstrukturen gelesen werden, dann besonders, wenn sich in ihnen in sich zwei gegensätzliche Prinzipien der Sprache virulent vermischen, das strukturelle und das performative.