Read/Write

Methodik des Codes im künstlerischen Media Hack

„Programmieren ist vorbei.“ (Hans Bernhard)

Um die Jahrtausendwende gab es mit der Blütezeit der Net-, Code- und Software Art eine diese begleitende intensive theoretische Auseinandersetzung mit Programmiertexten und -codes. Seither hat die Durchdringung des Alltags, der Verwaltung, der Arbeitswelt und der Kommunikation durch softwaregesteuerte Prozesse massiv zugenommen. Gleichzeitig stelle ich aber fest, dass sowohl die künstlerische als auch die theoretische Beschäftigung mit der Basistechnologie unserer Mediengesellschaft diese Entwicklung nicht spiegeln. Im Gegenteil: 2003 hatte Matthew Fuller eine Kritik von Software gefordert und noch 2008 fragt er im Vorwort zu seiner Sammlung von Essays zu Software: „Where is the rest of that criticism?“ (Fuller: 2008, S. 2). Warum ist es um Code und Software so still geworden?
Ich wollte diese Frage angehen, indem ich KünstlerInnen auf ihr persönliches, durch ihre künstlerische Arbeit erworbenes Wissen zu Code und Software befragte.[1] Im Workhsop wurde deutlich, dass Code bei den meisten KünstlerInnen nicht im Zentrum der Aufmerksamkeit steht, obwohl er durchaus ein Teil ihrer künstlerischen Praxis ist. Im Gespräch bildeten sich Media-Hacking und Narration als zentrale Themen heraus, wobei auffallend oft der Begriff der „Story“ fiel.
Inke Arns hat in einem Katalogbeitrag für ubermorgen.com narrative Strategien als Kern von Media-Hacks beschrieben (Arns: 2009). Narration ermöglicht es nach Arns, sich in einer Welt, die sich zunehmend von der direkten Erfahrung zurückzieht, Sinn herzustellen und Leute zu mobilisieren. Sie nimmt dabei weniger Bezug auf die Net-Art, sondern bezieht sich vor allem auf die (durchaus damit verbundene) Tradition der aktivistischen künstlerischen Praxis. Der vorliegende Text möchte dagegen einen Versuch wagen, einen Übergang von einer Praxis des Codes zu einer „Praxis der Story“ herzustellen.
Anhand der Gespräche während des Workshops möchte ich herausarbeiten, wie die künstlerische Praxis mit Code sich zu einer narrativ/performativen Praxis mit Medien entwickeln konnte bzw. musste. Über die Erfahrungen von KünstlerInnen soll hier über den auf den ersten Blick überraschenden Zusammenhang von Code und Narration nachgedacht werden.

---

[1] Der Workshop fand am 18.5.2012 an der Zürcher Hochschule der Künste, Zürich statt. Es nahmen Teil: (Alexander Tuchacek (knowbotic research), Carmen Weisskopf und Domagoj Smoljo (!Mediengruppe Bitnik), Hans Bernhard (ubermorgen.com) und Johannes Auer.

Handlungsraum Code

Software als die Daseinsform von Code ist und war schon immer eine vielschichtige Angelegenheit: zuunterst liegt der Maschinencode, darüber liegen die Ebenen der Interpreter, Compiler, Programmiersprachen, Protokolle, Interfaces und Betriebssysteme. Code ist auf all diesen Ebenen präsent in seiner der Ebene entsprechenden Form und Funktion[2]. Was im Bereich der Personal-Computer spätestens seit der Einführung der ersten GUIs von Apple Anfang der 1980er-Jahre komplex ist, war im Internet bis zum Aufkommen des Web 2.0 nach der ersten Dotcom-Krise um die Nullerjahre verhältnismässig flach. Es gab eine Programmiersprache (HTML) und einen Interpreter (Browser). Code war für alle erreichbar – „geradezu paradiesische Zustände" nannte Johannes Auer die Zeit der blühenden Netzkunst.

Die Weiterentwicklung des Netzes hat dazu geführt, dass heute primär Plattformen benutzt werden. Facebook, Blogs, Plattformen wie YouTube, FlickR, Twitter etc ermöglichen das Publizieren von Texten, Bildern, Videos und Nachrichten im Netz, ohne mit Code in Berührung zu kommen. Die Welt des Codes hat sich auch im Internet von den Oberflächen zurückgezogen – oder anders ausgedrückt: die Ebene des Codes wird von weiteren Code-Ebenen (Plattformen, Apps) überlagert. Dieser Code ist oft nicht mehr offen und zugänglich, sondern Eigentum der jeweiligen Plattformbetreiber und somit unerreichbar. Dadurch wird der offene, zugängliche Code durch eine darauf aufsetzende weitere Ebene wieder geschlossen.
Damit ist gemeint, dass Code nicht mehr gelesen und (mit-)geschrieben werden kann, dass seine Entwicklung sich den Usern entzieht. Der Begriff „Code" meint in diesem Zusammenhang Code-als-Text, also Code in seiner sprachlichen Form, wie ihn Programmierer schreiben, wie ihn an Programmiersprachen geschulte Menschen lesen, verstehen und ggf. weiterschreiben können. Code-als-Text wurde im Kontext der Netz- und Softwarekunst ausführlich diskutiert (siehe Arns, Cramer).
Das zunehmende „Layering" von Code und die damit verbundene Schliessung von zugänglicher und freier Software hat den Zugang zum Handlungsraum Code für KünstlerInnen schwierig und unattraktiv gemacht.[3] Hans Bernhard drückte es folgendermassen aus: „der Code ist zuunterst, da kannst du was machen, aber er ist fast unerreichbar."

---

[2] Laut N. Katherine Hayles bringt diese Schichtung von Code und ihre Beziehungen untereinander den "flickering signifier" hervor, zeitbasierenden Bedeutungsketten, die erst durch die Verschachtelung von Zeichen und Bezeichnetem auf den unterschieldichen Code-Ebenen entstehen. Für Hayles sind die flickering signifier das signifikante Neue an der Weltsicht des Codes und strukturierend für all unsere Erfahrungen mit Code.

[3] Was nicht heisst, dass diese Arbeit nicht mehr stattfindet. Ein Beispiel ist die „suicide machine" von Gordan Savicic, Walter Langelaar und Danya Vasiliev vom niederländischen Medienlabor WORM: http://suicidemachine.org/. Gordan Savicic war ebenfalls zum Workshop eingeladen, konnte aber leider nicht teilnehmen.

Running Code

Dass Code uns nicht mehr als Text zugänglich ist, bedeutet jedoch nicht, dass Code an sich aus unserem Erfahrungs- und Gestaltungsbereich verschwunden ist. Aber er erscheint uns in einer anderen Form, und erfordert folglich auch einen anderen Umgang. Welcher Begriff könnte sich eignen, diese Erscheinungsform von Code heute zu benennen?
John Cayley, ein Künstler und Netzliterat, veröffentlichte 2002 ein Essay, in dem er die Utopie der Code-Sprache-Transparenz in der Theoretisierung der Netz- und Softwarekunst kritisierte (Cayley: 2002). Er monierte, dass der Code mit dem Text verwechselt werde und dabei vergessen werde, dass zum Code auch immer seine Ausführung gehöre. Damit hat er die doppelseitige Daseinsform von Code benannt: einerseits den Code-als-Text, andererseits den laufenden Code. Laut Cayley besteht Code aus beiden Seiten, und diese sind nicht deckungsgleich: der Code ist nicht (nur) der Text. Im Gegenteil: Code-als-Text

> „has ceased to be operative or even potentially operative. It is 'broken' in the now familiar programmer's jargon. The breakdown of its operations eliminates one aspect of its proposed aesthetic value and allure, its native performative efficacy [...]." (Cayley: 2002)

Cayley bezeichnet hier Code-als-Text als "broken code". Im Gegensatz dazu könnte man den laufenden Code als „Running Code" bezeichnen. Im Gespräch wurde von Alexander Tuchacek Running Code als aktueller Codebegriff vorgeschlagen. Running Code ist der vom Computer prozessierte Code während seiner Laufzeit. Es ist der Code, der seine technische Performativität vollzieht. Running Code ist das Gegenteil von Code-als-Text. Running Code bedeutet Kontrollverlust: er ist nicht mehr beeinflussbar, er ist nicht mehr lesbar, nicht schreibbar, nicht adaptierbar, nicht kopierbar, er ist auch nicht verhandelbar. Es ist der Code welcher handelt.
Unsere Begegnungen mit Code im Alltag sind Begegnungen mit Running Code: die Plattformen des Web2.0 und die Apps der Smartphones gehören zum Running Code, aber auch die automatisierten Handelssysteme der elektronischen Börsen. Code-als-Text ist (wieder) zunehmend den Spezialisten vorbehalten (wobei dazu nicht nur die Informatiker der Softwarekonzerne zählen, sondern auch die Hacker und Nerds, die Aktivisten der Open Source Software, die Amateure).
Code-als-Text war der leitende Codebegriff der Netz- und Softwarekunst, zu einer Zeit, als Code im Internet dank dem flachen Layering relativ einfach zugänglich war. Da diese Zugänglichkeit heute wegen der technologischen Weiterentwicklung des Internets nicht mehr gegeben ist, haben sich zwangsläufig auch der Codebegriff und die künstlerischen Strategien im Umgang mit Code verändert.

Wenn Running Code uns als Tatsache begegnet, wie können wir diese einordnen? Matthew

Fuller beschreibt Software als etwas, was auf vielen Ebenen existiert, und das sich mit vielen verschiedenen Dingen verbindet.

> „And it is this paradox, the ability to mix the formalized with the more messy – non-mathematical formalisms, linguistic, and visual objects and codes, events occurring at every scale from the ecological to the erotic and political – which gives computation its powerful effects, and which folds back into software in its existence as culture.“ (Fuller, 2008, S. 5/6)

Auch Adrian MacKenzie begründet in seiner Untersuchung zu Linux den Status von Software als kulturelles Objekt in den Verbindungen, die diese durchqueren:

> "Within technological cultures, operating systems and server software constitute just such things which we might term 'culture objects' by virtue of the density of the mediations and relationality that run through them and texture them." (MacKenzie: 2005, S. ?)

Fuller und MacKenzie betonen beide, dass Software (und dies kann auf Running Code übertragen werden) mit der Welt eng verbunden ist und dass diese Verbindungen die Mächtigkeit von Software begründen und ihren Status als kulturelles Objekt konstituieren. Ist Code in der gegenwärtigen Form von Running Code als Handlungsraum unzugänglich geworden, so verbinden ihn doch seine Runtime-, Herstellungs-, Distributions- oder Gebrauchskontexte mit der Welt, den Menschen und ihrem Handeln. Diese Verbindungen sind im Gegensatz zum Code-als-Text für eine künstlerische Praxis zugänglich.

Die Story

Wie nutzen KünstlerInnen die Verbindungen zum Running Code als künstlerisches Handlungsfeld? Hans Bernhard hat im Gespräch formuliert:

> „Code zu beherrschen ist heute unmöglich, ihn nur schon zu verstehen ist unmöglich. Aber wir können ihn benutzen.“

Wie ist das gemeint? Betrachten wir die Arbeit *[v]ote-auction*[4] von ubermorgen.com (Hans Bernhard und Lizvlx). Unter dem Namen vote-auction.net setzte die Künstlergruppe eine Versteigerungsplattform auf, über die das amerikanische Stimmrecht für die Präsidentschaftswahlen 2000 (G.W. Bush vs. Al Gore) an den Höchstbietenden verkauft werden konnte. Eine Statistik der aktuellen Gebotslage besagte: Durchschnittlich $ 33.33 pro Stimme in Colorado (8 Angebote), in Texas nur $ 6.45 (bei 32 Angeboten). Die Plattform mit dem Slogan„Bringing democracy and capitalism closer together“ warf mitten im Wahlkampf grosse Wellen: acht US-Bundesstaaten klagten gegen *[v]ote-Auction* wegen illegalem Stimmenhandel, das FBI und die NSA wurden eingeschaltet, um die Integrität der Wahl

4 http://www.vote-auction.net/

sicherzustellen, über 2500 Pressemeldungen wurden in Print, online und im Fernsehen gezählt. CNN widmete eine Ausgabe von „burden of proof“ (ein Sendeformat zu Rechtsfragen) und führte ein denkwürdiges Interview mit dem Künstler und zugeschalteten Experten. Schliesslich mussten alle Klagen zurückgezogen werden, denn die Versteigerungsplattform war ein simples HTML-Dummy, das die versprochenen Funktionalitäten nur simuliert hatte. Die Statistik war fikitv, alle Zahlen frei erfunden, es konnten sich weder Verkaufswillig noch Kaufinteressenten registrieren, kurz: der Seite konnte nicht einmal die Absicht zum Stimmenhandel nachgewiesen werden.
Diese Arbeit beinhaltet zwar Code in Form einer Webseite. Aber dieser Code wird durch seine offen eingestandene Disfunktionalität aufgelöst, er ist vielmehr die Behauptung eines Codes. Hans Bernhard meinte zu diesem Verzicht auf den Code: „die Story ist so gut, da brauchst den Code gar nicht mehr.“
Was aber ist die Story, die es erlaubt, auf den Code zu verzichten? Mit Story ist nicht eine Erzählung im herkömmlichen Sinne gemeint. Die Künstlergruppe fungiert weder als Erzähler noch als Autor. Hans Bernhard bezeichnet seine Rolle als „Spin Doctor“, als jemand, der den Ereignissen durch gezielte Intervention einen weiteren Dreh versetzen kann.
Das Ziel dieser Arbeitsweise sind also Ereignisse, die nicht erzählt, sondern in Gang gesetzt und gehalten werden. Die Story bezeichnet ein performatives Setting, mit dem Ereignisse erzeugt werden. Das Setting zeichnet sich durch ein gekonntes Arrangement von Kontexten aus: das radikale Aufeinanderbeziehen von Demokratie und Kapitalismus in Form einer Dienstleistung zum Zeitpunkt der Präsidentschaftswahlen. Diese Kombination von Kontexten ermöglicht es, den Code komplett zu vernachlässigen. Die performative Kraft geht nicht mehr vom Code aus, sondern von den Kontexten, die ihn begleiten.

Ein weiteres Beispiel: Für ihre Arbeit *opera calling* installierte die Mediengruppe !Bitnik einfache Wanzen im Zuhörerraum der Oper in Zürich. Über eine programmierte Schaltanlage wurde die Aufführung live an zufällig ausgewählte Telefonnummern des zürcher Telefonnetzes übertragen. Die Zuhörer hörten eine automatische Ansage und wurden dann auf die Live-Übertragung umgeschaltet. Die Künstler schnitten die Telefonate zu Dokumentationszwecken mit. Da die Übertragung kein Broadcast war, sondern in der Tradition der Telefonabonnemente aus der Gründerzeit des Telefons eine 1:1 Übertragung, war das Publikum dieser Aktion relativ klein. Die Sprengkraft der Arbeit erhielt sie erst, nachdem die Presse über eine Ausstellung darauf aufmerksam wurde. Das Opernhaus suchte verzweifelt die Wanzen und reichte Klage ein, die Presse reagierte begeistert und eine Debatte über das Verhältnis von Subventionssumme und Zielpublikum setzte ein. Auch hier wurde durch ein Setting von Kontexten ein Ereignis performativ hervorgerufen. Bei *opera calling* sind die Kontexte zweier Medien kontrovers aufeinander bezogen: die Aufführungsformat der Oper (des Theaters) und das Distributionsformat des Telefons.

Media Hacking

Beide aufgeführten Arbeiten werden gemeinhin als Media Hacks bezeichnet. Media-Hacking in künstlerischer Absicht bedeutet, die Eigengesetzlichkeiten von Massenmedien so zu nutzen, dass ein „Experiment von innen“ (Ausdruck: Hans Bernhard) sich selbstständig in einem Zusammenspiel der beteiligten Akteure entwickelt. Dafür entwickeln die KünstlerInnen ein performatives Setting, welches sie als Story bezeichnen. Performativ meint hier ein Wirken, das wirklichkeitskonstruierend und selbstreferentiell ist. Diese Definition bezieht sich auf die Sprechakttheorie von Austin, welcher in seiner bahnbrechenden Vorlesung in Harvard 1955 erstmals auf die Fähigkeit der Sprache hinwies, Dinge nicht nur zu beschreiben sondern auch zu vollziehen, d.h. aussersprachlich wirksam zu sein. Das damit eröffnete Feld des Performativen ist gross. Judith Butler beschrieb die geschlechtliche (und damit soziale) Identität als in performativen Prozessen herausgebildet (Butler: 2003 [1990]). Erika Fischer-Lichte erweiterte die Sprechakttheorie in ihrer Analyse der Performancekunst und des Theaters, in der sie den Wandel vom Werk zur Aufführung in den Künsten seit den 1960er Jahre untersucht (Fischer-Lichte: 2004). Das Performative ist mit Erika Fischer-Lichte gesprochen, kein Zeichen für Bedeutung, sondern es bringt diese selber in engem Verbund mit der Materialität hervor. Auch Inke Arns Analyse von Programmtexten bezieht sich auf Austins Theorie der Sprechakte (Arns: 2005).

Nach dem Stellenwert der Netzkunst in ihrem künstlerischen Universum befragt, bezeichneten die meisten Gesprächsteilnehmer die Netzkunst als die „Antike“, als historische Referenz (unter anderen) ihrer eigenen künstlerischen Arbeit. Carmen Weisskopf sagte: „Wir sind später gekommen. Was wir übernommen haben, ist das Read/Write.“ Read/Write ist die Methodik des Lesens und Umschreibens von Code-als-Text, und bedingt das Wissen um die Performativität dieser Texte. Ein Programm-Code umzuschreiben heisst nicht nur, danach einen anderen (Programm-)Text zu haben, sondern es bedeutet auch, ein anderes Verhalten des Programmes zu bekommen. Die KünstlerInnen haben diese Strategie vom Programmcode auf kulturelle, soziale und politische Codes übertragen. Die Wirklichkeit, wie sie uns in Form von Institutionen und Zusammenhängen entgegentritt, wird so zu einem performatives Feld, das den Handlungsraum der KünstlerInnen radikal erweitert.
Mit der Methode des Read/Write transferieren die KünstlerInnen das Wissen um performative Zusammenhänge aus dem technologischen Setting in die Welt und findet diese Zusammenhänge dort überall und der Methodik des Read/Write ebenso zugänglich wie beim Programmieren.[5] Der Begriff des Narrativs, welchen Inke Arns im Zusammenhang mit Media Hacking gebraucht (Arns: 2009), ist also nicht deskriptiv, sondern performativ zu verstehen.

[5] Florian Cramer hat eine Genealogie des performativen Sprechens geschrieben, welche er als Pretext zur Programmierung beschreibt: Cramer: 2003.

Das „Storytelling in the Information Age“ (Paraphrase auf den Titel von Arns Katalogbeitrag) erlaubt es nicht nur, die Welt zu beschreiben – es ist vielmehr eine Strategie, an der Welt mitzuschreiben.

Programmieren die KünstlerInnen also die Realität? Ist die Wirklichkeit eine Art von Running Code? Aus der Perspektive der Methodik kann man dies bejahen. Bezieht man aber die Rolle der KünstlerInnen in die Betrachtung mit ein, stellt man fest, dass diese nicht mehr mit Programmierenden vergleichbar ist. Der Programmierer unterscheidet sich fundamental von seinem Programm. Zwischen Code schreiben und ausführen steht immer ein Wechsel des Mediums, eine Übersetzungsleistung. Code wird kompiliert bevor er ausgeführt wird, dies markiert den Wechsel vom Code-als-Text zum Running Code. Diese klare Trennung der Ebene des Schreibens und des Ausführens gibt es in der beschriebenen künstlerischen Praxis nicht mehr. Die KünstlerInnen sind Teil des Settings, sie gehören dem Gefüge an, woran sie mitschreiben und sie können sich den Wirkungen ihres Tuns nicht entziehen.[6] In der beschriebenen künstlerischen Praxis verschränken sich die Ebenen des Schreibens und des Ausführens miteinander und wirken aufeinander zurück.
Mit dem beschriebenen Methodentransfer hat sich – und das ist vielleicht das wichtigste daran – der Ort des Performativen verschoben. Das Performative wird nicht mehr als eine Eigenschaft des Materials (des Codes) behandelt, sondern wandelt sich zum Kern einer künstlerischen Strategie, welche die Gegenwart als performativ hervorgebracht zum Thema hat. Dies verändert den Blick auf alles: in der Story vollzieht sich Wirklichkeit als Fiktion.

Referenzen:

Arns, Inke: *„Code as performative Speech Act“*, in: Artnodes, e-journal on art, science and technology, 2005/7

Dies.: „Storytellers of the Information Age - on the Role of Narrative in ubermorgen.com's Work“, in: Quaranta Dominic (Hg.), *ubermorgen.com*, Brescia 2009

Austin, John L.: *„Zur Theorie der Sprechakte (How to do Things with Words)“*, Leipzig 1972 (1962)

Butler, Judith: *„Das Unbehagen der Geschlechter“*, Frankfurt a.M. 2003 (1990)

Cayley, John: *„The Code is not the Text (unless it is the Text)“*, in: electronic book review, Sept. 2002

Cramer, Florian: „Concepts, Notations, Software, Art“, in: Guriunova, Olga / Shulgin Alexej (Hg.), *readme 1.2*, Moskau 2002

[6] Genauso wenig wie sich die weiteren, meist unfreiwilligen Akteure sich auf einen ästhetische Position zurückziehen können. Dies weil die Aktionen nicht institutionell gerahmt sind. Das fehlen der Information „das ist Kunst“ macht einen grossen Anteil der Mobilisierungskraft der Aktionen aus.

Ders.: *„Words made Flesh“*, Rotterdam 2003, online: http://pzwart.wdka.hro.nl/mdr/research/fcramer/wordsmadeflesh/

Ders.: „Exe.[cut]able Statements: Das Drängen der Codes an die Nutzeroberflächen“, in: Gerfried Stocker, Christine Schöpf (Hg.), *Code – the Language of our Times*, Linz 2003

Fischer-Lichte, Erika: *„Ästhetik des Performativen“*, Frankfurt a.M. 2004

Fuller, Matthew: *„Behind the Blip: Essays on the Culture of Software“*, New York 2003

Ders.: *„Software Studies – a Lexikon“*, Cambridge 2008

Hayles, N. Katherine: *„My Mother Was a Computer - Digital Subjects and Literary Texts“*, Chicago 2005

MacKenzie, Adrian: *„The Performativity of Code: Software and Cultures of Circulation“*, in: Theory, Culture and Society, 2005/22 Nr. 1

Alle Zitate der KünstlerInnen sind dem Mitschnitt der Gespräche im Rahmen des Workshops vom 18.5.2012 an der Zürcher Hochschule der Künste entnommen.

GesprächsteilnehmerInnen:

Alexander Tuchacek (knowbotic research): http://krcf.org
Domagoj Smoljo und Carmen Weisskopf (!Mediengruppe Bitnik): http://www.bitnik.org
Hans Bernhard (ubermorgen.com): http://www.ubermorgen.com , http://www.hansbernhard.com
Johannes Auer: http://auer.netzliteratur.net/

Shusha Niederberger, 2012