

konkret digital

Interview mit Johannes Auer über Konkrete Poesie und Netzliteratur (2011)

Erscheint in: Concrete Poetry: An International Perspective. Edited by Claus Clüver and Marina Corrêa. Das Interview führte **Dr. Martina Pfeiler**, Institut für Anglistik und Amerikanistik, TU Dortmund

Wann sind Sie zum ersten Mal mit der Konkreten Poesie in Berührung gekommen? Ich denke hier an ihren Einblick in die Stuttgarter Schule sowie ihren kinetischen Apfel "worm applepie for doehl".

Mit Konkreter Poesie im Bezug auf meine eigene künstlerische Arbeit habe ich mich intensiver erst während meiner Zusammenarbeit mit Reinhard Döhl auseinandergesetzt. Ich habe Reinhard Döhl 1994 auf dem Stuttgarter Symposium zu Max Bense kennen gelernt und ab 1996 mit ihm begonnen, netzliterarisch zu experimentieren. 1997, als Geburtstagsgeschenk, habe ich für ihn seinen berühmten *apfel* als Computeranimation geconvert. Vielleicht interessant in diesem Zusammenhang, dass ich im Rückblick diesen *worm applepie for doehl* durchaus kritisch sehe. Während Döhls *apfel* von 1965 den erzeugenden Code reflektiert, vermeidet dies meine Animation, sie verbirgt den erzeugenden Programmcode. Deshalb habe ich 2004 das Apfelspiel mit der Codezeile `$wurm = ($apfel>0) ? 1 : 0;` weitergetrieben. In Worten ausgedrückt besagt dieser Code genau das, was die Apfelanimation vorführt: *Ist der Apfel größer Null, is(s)t der Wurm. Ansonsten is(s)t er nicht.* Diese literarische Codezeile ist ein Codework, ist ausführbarer Programmcode.

Auch in der Arbeit *G-Linie HTML*¹, zu der ich 2008 von der Poesiewerkstatt Berlin beauftragt wurde als digitale Auseinandersetzung mit Eugen Gomringer, wird vordergründig der Programmcode als erzeugendes Material des Bildschirmereignisses thematisiert. Um nicht missverstanden zu werden, ich bin nicht der puristischen Ansicht, dass wahre Netzliteraten nur noch mit dem Code selbst dichten sollten. Ich bin aber andererseits mittlerweile sehr wohl der Ansicht, dass im Oberflächentext der erzeugende Subtext, also der Algorithmus, in irgendeiner Form begreifbar werden muss. Ich denke, wir werden an anderer Stelle noch darauf zu sprechen kommen.

In der Auseinandersetzung mit der Stuttgarter Schule ist für mich von besonderem Interesse die Unterscheidung von Max Bense zwischen künstlicher und natürlicher Poesie, die Bense 1962 unter dem Eindruck von Theo Lutz "Stochastischen Texten" traf². Ich habe Theo Lutz Arbeit 2004 nachprogrammiert und ab 2005 weiterentwickelt zu den Performanceprojekten *Free Lutz!*³ und *searchLutz*⁴, bei denen u.a. die Frage im

Zentrum steht, wie man aus künstlicher Poesie natürliche Poesie schaffen kann - gerade unter Berücksichtigung der medialen Bedingungen.

In Ihrer Einleitung zum Gespräch „Konkret Digital Döhl“, das Sie mit Peter Weibel und Michael Lentz im Stuttgarter Literaturhaus im Jahr 2009 geführt haben (http://www.literatur-und-strom.de/3/index.html#weibel_lentz_auer), betonen Sie, dass es für einen Autor der Konkreten Poesie wie Reinhard Döhl nicht überraschend war, dass er eine Verbindung mit Netzliteratur eingegangen ist. Ist es umgekehrt ebenso gewiss, dass sich (ggf. Ihnen bekannte) NetzkünstlerInnen mit Konkreter Poesie in Buchform beschäftigen oder früher beschäftigt haben? Sehen Sie sich als Teil einer Gruppe von KünstlerInnen?

Die frühen literarischen Computereperimente der Stuttgarter Schule und deren theoretische Reflexion waren in der Tat ein wichtiger Anknüpfungspunkt für die Auseinandersetzung von Reinhard Döhl mit der Netzliteratur. Auch konnten Arbeiten von ihm aus den 60er Jahren leicht für das Netz aktualisiert werden - beispielsweise das *Buch Gertrud*⁵ und sein permutativer Text *Der Tod eines Fauns*⁶. Ist doch gerade die Permutation ein Text mit geringer Autorpräsenz, der auf der Ausführung einer Regel basiert und deshalb geradezu prädestiniert erscheint für eine digitale Realisation. Hinzu kam ein internationales Autorennetz, das Reinhard Döhl für unsere ersten literarischen Netzliteraturexperimente Anfang 1996 aktivieren konnte. An dialogischen Gemeinschaftsarbeiten wie *HHH - Hommage à Helmut Heißenbüttel*⁷ beteiligten sich u.a. Ernst Jandl und Friedericke Mayröcker, Haroldo de Campos, Pierre und Ilse Garnier, Bohumila Grögerova und Josef Hisal.

Auch einer Arbeit wie Eugen Gomringers *3 variationen zu 'kein fehler im system'* von 1969 liegt sozusagen ein ausgeführter Algorithmus zu Grunde, demgemäß sich der Buchstabe f immer um eine Position nach rechts verschiebt.

Die Frage, ob und in welcher Intensität sich Netzliteraten generell mit Konkreter Poesie in Buchform beschäftigt haben, kann ich nicht beantworten. Festzustellen ist aber, dass es gerade in der ASCII-Art zu einer Art Wiedererfindung der Konkreten Poesie im digitalen Medium kommt. Die Kommandozeile war zunächst bei Betriebssystemen das einzige Eingabeinterface. Der Computer wurde dort über Textbefehle gesteuert.

Von Anfang an wurde die Kommandozeile aber auch benutzt, um ganz anderes zu tun. Es wurde sehr schnell versucht mit den Mitteln des ASCII-Codes⁸ Bilder zu erzeugen. Wir kennen das heute noch alle durch die Textsmileys in E-Mails. Diese, wie man sagen könnte, Subversion der Kommandozeile hatte zunächst einen ganz praktischen Grund. Beim Drucken waren anfänglich nur ASCII-Zeichen erlaubt. Wollte man also ein Bild ausdrucken, musste man es aus Buchstaben gestalten. Das Genre der ASCII-Art ist eine entscheidende Basis für die oft behauptete Analogie von Konkreter und digitaler Poesie. In wieweit sich nun aber Netzkünstler wie Joan Heemskerk und Dirk Paesmans von JODI oder Walter van der Crujisen, Luka Frelj und Vuk Cosic vom ASCII Art Ensemble dezidiert mit Konkreter Poesie auseinandergesetzt haben, entzieht sich meiner Kenntnis.

Und zum letzten Teil der Frage: in der Tat sehe ich mich einer Künstlergruppe zugehörig, zu der ich alle rechne, die sich mit Netzliteratur und Netzkunst zu den ihr eigenen medialen Bedingungen beschäftigen: Textbasierung digitaler Technologie, algorithmische Steuerung und interaktive Architektur.

Sie sprachen vorhin den expliziten Verweis erzeugenden Codes auf sich selbst an.

Welche Unterscheidung machen Sie in Hinblick auf visuelle oder akustische Dimension ihrer Werke und dem Programmiercode?

Bei Codeworks fällt Phäno- und Genotext in eins, wie Inke Arns ausgeführt hat⁹. "Sagen" (Genotext) und "Tun" (Phänotext) fallen zusammen. Wenn in der Kommandozeile des Computers "kill" eingegeben wird, geschieht genau das. Codeworks (= Dichtungen mit Programmcode) erscheinen so poetologisch zunächst reizvoll, sind ausgeführt meines Erachtens aber oft eher unbefriedigend. Entweder sie bleiben im kunsthandwerklichen Bereich von Hacker-Spielereien stecken (wie beispielsweise die meisten Perl-Gedichte) oder müssen in geglückteren Beispielen, wie Jaromils *ASCII Forkbomb*, metaphorisch stark aufgeladen werden. So spicht Jaromil von einer viralen, verfluchten Poesie, "die gegen jene (revoltiert), die das Netz als einen sicheren und bourgeoisen Ort zu verkaufen versuchen"¹⁰.

In der zweiten Gestalt werden Codeworks zu "broken Code" sind also nicht mehr vom Computer ausführbar, werden damit zu Code dekorierten Privatsprachen, wie beispielsweise bei der australischen Netzliteratin MEZ, und nähern sich, überspitzt gesagt, genialischer Poesie an. "Hier sitz ich, forme ein Codemensch

Sprachenamalgam nach meinem Willen" könnte man dazu die berühmten programmatischen Zeilen aus Goethes Prometheus vielleicht umschreiben. Zur Frage: In meinen jüngeren Arbeiten (*searchLutz*, *searchSongs*¹¹, *searchSonata 181*¹²) ist der Code, der Algorithmus leicht verständlich, kann also in der Performance vom Betrachter mitvollzogen werden. Bei *searchLutz* sind es die *stochastischen Texte* von Theo Lutz, ein Algorithmus, der nach einem simplen Regelbausatz Sätze generiert. Die *searchSongs* nutzten einen einfachen Code mit dem beispielsweise schon Johann Sebastian Bach gearbeitet hatte, nämlich die Regel, dass Buchstaben Tönen der Notenscala entsprechen (also im Beispiel von Bach B-A-C-H). Und in die *searchSonata 181* ist ein Passwortalgorithmus eingearbeitet, der die simple Regel befolgt, aus den gegebenen Wörtern zufällige Buchstabenfolgen zu montieren.

In der Interaktion kann der Besucher der Performance also, indem er Wörter eingibt, den Algorithmus mitdenken und das Ergebnis des generierten Textes mitplanen. Er kann im generierten Phänotext, den Genotext mitvollziehen. Der Code ist also nicht unsichtbar, nicht "transparent", wie sonst in den meisten Computeranwendungen.

Entscheidend ist nun aber, dass die Kalküle des teilnehmenden Besuchers nicht im Algorithmus stecken bleiben sondern an ein "menschliches Interface", einen Sprecher oder Musiker übergeben werden. Dieser reagiert als autonomer und interpretativer Dialogpartner und nicht als ausführender Rechenknecht. Der algorithmisch generierte Text wird also akustisch und theatralisch gestaltet. Wird zur Partitur für eine visuell-akustische Live-Aufführung.

In den USA war bis in die 1970er Jahre und darüber hinaus ein großer Widerstand innerhalb traditioneller kultureller Einrichtungen und insbesondere auch an Universitäten gegenüber jenen AutorInnen zu spüren, die versucht haben, das Buch und später den Computer als intermedialen Raum bzw. Werkzeug experimentell auszuloten. Wie sehen Sie vergleichsweise die Situation für NetzliteratInnen in Deutschland?

Die Situation für die Netzliteratur in Deutschland war in der Anfangszeit ab 1996 genau gegenteilig. Gerade die universitäre Forschung stürzte sich mit postmodern geschultem Blick begeistert auf Hypertext und Hyperfiction, erkannte darin eine sichtbar gemachte Intertextualität und einen Leser, der im Lektüreprozess zum Autor wird. Zeitweilig hatte man fast den Eindruck, dass es mehr Forschungsarbeiten über hypertextuelles

Schreiben gab, als Werke selbst. Wer schreibt den Hypertext, der Autor oder der Leser? Das war die rhetorisch gestellte Frage. Argumentiert wurde folgendermaßen: im Internet sei jeder Leser gleichzeitig Autor, da er über die Links, die er anklicke, die Textgestalt bestimme. Er collagiere also beim Lesen seinen Text, stelle bei der Lektüre den Text her. „In cyberspace“, so Benjamin Whooley, "everyone is an author, which means no one is an author: the distinction from the reader disappears. Exit author..."¹³. Der Wreader¹⁴ also tatsächlich der Todesklick für den Autor? Diese Frage wurde u.a. von Uwe Wirth schon früh zurecht als Ideologie entlarvt: „In dem Maße, in dem Hypertexte auf eine Struktur, bzw. auf eine interne Kohärenz verzichten, um sich ganz den Entscheidungen des Lesers zu öffnen, verwischt die Grenze zwischen Interpretation und Gebrauch. Ein total offener Hypertext ist daher völlig uninterpretierbar.“¹⁵ Oder anders ausgedrückt: Ein offener Hypertext ist sinnlos. Ein sinnvoll zu lesender Hypertext braucht immer den Autor, zumindest als Regisseur, der die Lektüremöglichkeiten beschränkt und damit in gewissem Maße kontrolliert. Parallel zur universitären Hypertext Begeisterung entwickelte sich ein Medienhype ums Internet, der auch Netzkunst und Netzliteratur einschloss.

Dieser Rausch in der zweiten Hälfte 1990er Jahre, diese anfängliche euphorische Überschätzung des Internets, auf das unglaubliche gesellschaftliche, politische, künstlerische und dann vor allem auch wirtschaftliche Phantasien projiziert wurden, dieser Rausch endete mit dem Zusammenbruch der Dotcom-Blase im Jahr 2000 zunächst mit einem großen Kater. Dennoch wurde, einer geradezu olympischen Logik des Hypes "höher, schneller, weiter" folgend, danach sofort wieder versucht, das Netz unkritisch zu überhöhen. Eines der letzteren Beispiel ist die erst langsam abebbende Social Media Euphorie. Dieser Logik des Hypes konnte weder Netzliteratur noch Netzkunst auf Dauer gerecht werden. Daher regiert bei einigen Interpreten und Kuratoren mittlerweile eher Ernüchterung und der anfänglich so heftig bejubelte Tod des Autors im Hypertext wandelte sich zum Abgesang auf Netzkunst und Netzliteratur und zur Flucht in "Game Studies", zur Flucht in die angesagtere Computerspielwelt, die, so die These, die Netzliteratur evolutionär beerbe¹⁶. Die Netzkunst und Netzliteratur hat den Hype überstanden und übersteht auch, bei guter Konstitution, spielerische Missverständnisse.

Ein Computer ist keine Multimedia- sondern eine Textmaschine. Multimedia ist immer nur die Oberfläche, das Interface wie der Bildschirm oder die Lautsprecher. Immer,

wenn es beim Computer Multimedia wird, wird es analog. Digitale Technologie besteht auf der symbolischen Ebene aus Schichten von Texten: Computerprogrammen, Protokollen, ja auch der 01 Code ist wie das Morsealphabet ein Text. Der Schriftsteller William Gibson hat schon 1979 den Begriff Cyberspace in einer Erzählung erfunden. William Gibson sagte kürzlich in einem Interview "Cyberspace ist hier, jetzt, heute, da, wo wir gerade sind."¹⁷ Und meint damit die textuelle Kommunikation unserer digitalen Geräte, die für uns „transparent“ - also nicht wahrnehmbar - abläuft. Wenn also die digitale Technologie einen unsichtbaren Textraum, einen textuellen Cyberspace, um uns herum erzeugt und man der Kunst die Gestaltung von Wirklichkeit als Aufgabe zugesteht, dann hat gerade Netzkunst und Netzliteratur noch ein sehr weites Feld zu beackern.

Kommen wir nochmal zu "Free Lutz!" und "searchLutz" zurück aber auch zu Ihren neuesten Projekt mit Beat Suter und René Bauer, der "searchSonata 181" (<http://searchsonata.netzliteratur.net/fiwi/info.html>). Was macht Ihrer Erfahrung nach bei diesen Projekten die Faszination der Schaffung von "künstlicher Poesie" in "natürliche Poesie" sowohl für die Programmierer als auch die Performer aus? Was genau wird bei diesem Cross-Over "natürlich" gemacht?

"Poesie (ist) heute kein Transportmittel mehr für zumeist fragliche ethische Inhalte, kein Rechtfertigungsvehikel mehr für weltanschaulichen Unfug"¹⁸, schreiben 1964 Max Bense und Reinhard Döhl im Manifest der Stuttgarter Schule "Zur Lage" und konstatieren eine Tendenz der Poesie in Richtung einer "perfektionierten künstlichen Poesie". Im Umkehrschluss darf man folgern, dass gerade künstliche, also maschinell erzeugte Poesie, die ja nach Benses Definition "kein personales poetisches Bewußtsein mit seinen Erfahrungen, Erlebnissen, Gefühlen, Erinnerungen, Gedanken, Vorstellungen einer Einbildungskraft" haben kann also „keine präexistente Welt“¹⁹, sich dem menschlichen Missbrauch für "weltanschaulichen Unfug" entzieht und vielmehr einer "einer ästhetischen Negation gesellschaftlicher Zustände, zivilisatorischer Mängel"²⁰ dienen kann.

Heute hat sich meines Erachtens die Problemstellung umgekehrt. Computertechnologie wird, zugespitzt formuliert, oft nicht mehr emanzipatorisch eingesetzt, sondern beutet den Menschen zunehmend für ein Geschäftsmodell aus. Am Beispiel Google: nicht der geniale Algorithmus von Larry Page und Sergei Brin ist für die guten Suchergebnisse

verantwortlich, sondern die Tatsache, dass er die menschlichen Bewertungen und Wertschätzungen verarbeitet, die im Setzen eines Links zum Ausdruck kommt.

Baron Wolfgang von Kempelen begeisterte 1769 die Gesellschaft mit einem Schachroboter. Dieser sogenannte Schachtürke verblüffte seine menschlichen Gegner durch überragendes Schachspiel. Tatsächlich war das aber nicht einer genialen Mechanik geschuldet, sondern vielmehr der Tatsache, dass in diesem Automaten ein leibhaftiger kleinwüchsiger Schachmeister steckte. Ungefähr dieselbe Position nimmt der Mensch heute in Googles Algorithmus ein.²¹

Für mich ist nicht interessant, raffinierte Poesie-Generatoren zu programmieren. Vielmehr beschäftigt mich die Frage, wie bekomme ich den Menschen wieder authentisch in ein Setting mit künstlicher Poesie hinein?

Meine augenblickliche Antwort ist, dass das nur über eine Vermenschlichung des Ausgabeinterfaces gelingen kann, beispielsweise durch einen Sprecher, der – wie vorher schon ausgeführt – die algorithmisch generierten Texte interpretierend aufführt. Und somit mit seiner "präexistenten Welt" die künstliche Poesie in natürliche Poesie umwandelt.

Wie sehen Sie in Bezug auf Gomringer bei Ihrem Netzkunstwerk "G-Linie HTML", die interaktive Semantisierung von Raum? Welchen Stellenwert hat Gomringers Textgrundlage für Sie in diesem Werk?

Jeder Browser hat eine Funktion, die den Quelltext der aufgerufenen Webseite anzeigen kann. Der Internetnutzer kann also jeder Zeit sehen, wie die aufgerufene Webseite gebaut, programmiert ist.

Damit spielt die *G-Linie HTML*. Webseiten werden durch Markups, durch HTML-Tags gelayoutet. So umschließt und definiert das Tag-Paar `<p></p>` beispielsweise einen Absatz. Mit jedem ASCII-Editor können Webseiten gestaltet werden. In diesen Editoren (und nichts anderes ist auch die Quelltextanzeige des Browsers) können Zeilenumbrüche und Wortabstände vorgenommen werden, die im Browser aber erst dann sichtbar werden, wenn sie darüber hinaus mit den entsprechenden Tags ausgezeichnet sind. D.h. ein im Quelltext formatierter ASCII-Text ohne HTML-Tags erscheint im Browser unformatiert, als ununterbrochene Aneinanderreihung von Zeichen und Wörtern.

JODI hat in der Arbeit *%location*²² virtuos damit gespielt. Ruft man die Seite im Browser auf, sieht man nur eine blickende, unstrukturiert zusammenhängende Folge von Zeichen. Wechselt man jedoch in den Quelltext, dann sind dort ASCII-Grafiken zu sehen.²³

Genau darauf bezieht sich die *G-Linie HTML*, deren Untertitel *quelltext-hommage aah gomringer/jodi/la monte young* lautet. Im Browser bekommt der Betrachter nur eine horizontale Aneinanderreihung von Worten zu sehen. Wechselt er in die Quelltextansicht, so werden diese Worte quasi als ASCII-Art, als Gedichte von Eugen Gomringer dargestellt.

Des Weiteren kann der Betrachter auf der Browser-Ebene Wörter und Zeichen der Wortlinie ersetzen mit sofortiger Auswirkung auf den darunterliegenden Quelltext. Als Mitschreiber muss er also im Kopf eine Strategie entwickeln, den Subtext, die im Quelltext verborgenen Gedichte Gomringers, zu ändern. Dabei kann er die Struktur von Gomringers Gedichten - Zeilenumbrüche, Leerzeichen, Wortzahl - mit Geschick erweitern oder verringern, sie bleibt aber immer die Basis des Kalküls. Das verlangt sehr viel vom kollaborierenden Autor und weckt gelegentlich das Destruktive in ihm. Die *G-Line HTML* wird dann eher zu einem interaktiven Reload meiner Arbeit *Kill the Poem*²⁴ von 1997. Wie konstruktiv der Mitschreiber mein Handlungsangebot umsetzt, ist also ganz ihm überlassen. "Es kommt nicht wirklich darauf an, ob der Betrachter die Konzeption des Künstlers versteht (...). Wenn der Künstler sie einmal aus der Hand gegeben hat, hat er keine Kontrolle über die Art und Weise, wie ein Betrachter die Arbeit erfassen wird", schreibt Sol LeWitt 1967 in seinen "Paragrafen über konzeptuelle Kunst". Und die Konzeptkunst ist die dritte Referenz der *G-Linie HTML*. Von La Monte Young stammt die schöne Handlungsanweisung *Draw a straight line and follow it* (Composition 1960 No. 10). In der *G-Linie HTML* kann der Mitschreibende potentiell eine unendliche Buchstabenlinie konstruieren. Wirklich ausführen kann er dies aber, wie La Monte Youngs Anweisung, letztlich nur im Kopf.

Wie verorten Sie zum Beispiel "Free Lutz!" und "searchLutz!" im Bereich der internationalen E-Poetry Szene, bei denen in den letzten Jahren verstärkt die performative Live-Präsenz des Künstlers bzw. der Künstlerin in den Fokus rückt—etwa beim biennialen E-Poetry Festival in Paris 2007 (<http://epc.buffalo.edu/e-poetry/archive/>).

Bei Jörg Pieringers Auftritt und Aya Karpinskas Performance *Lala*²⁵, die auf der E-Poetry 2007 zu sehen waren, erkenne ich Parallelen zu meiner Arbeit: die Aufführung fügt dem elektronischen Text etwas Notwendiges hinzu. Der Unterschied liegt in den Akzentsetzungen. So steht bei Aya die Inszenierung mit Requisite im Vordergrund, während Jörg stark von der Programmierung her denkt, entwickelt und arbeitet. Ich dagegen akzentuiere die Interaktion, indem ich Netzkommunikation in Echtzeit miteinbeziehe (Wörter, die gerade in Suchmaschinen eingegeben werden) und das Publikum schreibend mitagieren lasse.

Da wir bei Veranstaltungen sind, würde ich gerne abschließend noch kurz auf Peter Weibels Ausstellung "Die algorithmische Revolution" am ZKM von 2004 und die international besetzte Sprachkunstreihe "3durch3"²⁶ von Friedrich W. Block zu sprechen kommen.

Im Ausstellungstext zur "algorithmischen Revolution"²⁷ stellt Weibel eine Verbindung her zwischen der Handlungsanweisung der Konzeptkunst an den Betrachter und der Rechenanweisung des Algorithmus an den Computer. Ersteres nennt er einen intuitiven Algorithmus. Diesen erkennt er insbesondere beim Fluxus, beispielsweise bei den *Event-Cards* George Brechts von 1961, mit denen dieser den Leser zu Aktionen aufforderte. Der exakte Algorithmus ist für Weibel hingegen die Rechenanweisung an den Computer, ist das Computerprogramm. Er sieht den exakten Algorithmus als präzise Fortsetzung des intuitiven Algorithmus.

Diesen Gedankengang weiterzuentwickeln zu einer Mischung von intuitivem und exaktem Algorithmus war eine weitere Inspirationsquelle für meine Netzperformances.

Bei seiner Veranstaltungsreihe "3durch3" lädt Friedrich W. Block jeweils drei Positionen der internationalen Sprachkunst gemeinsam ein. Visuelle, Laut- und digitale Poesie können so durchaus am selben Abend aufeinandertreffen. Diese erlebbare Verortung der digitalen Poesie im größeren sprachkünstlerischen Kontext finde ich wichtig und weiterführend.

- 1 <http://auer.netzliteratur.net/g-linie/>
- 2 Max Bense: Über natürliche und künstliche Poesie. In: Theorie der Texte. Eine Einführung in neuere Auffassungen und Methoden, Köln 1962
Online: http://www.stuttgarter-schule.de/natuerliche_und_kuenstliche_poesie.html
- 3 <http://freelutz.netzliteratur.net/>
- 4 <http://searchlutz.netzliteratur.net/>
- 5 <http://doehl.netzliteratur.net/gertrud/gertrud.htm>
- 6 <http://doehl.netzliteratur.net/faun/faun.htm>
- 7 http://auer.netzliteratur.net/hhh/h_h_h.htm
- 8 Der ASCII-Code umfasst die Zeichen des lateinischen Alphabets in Groß- und Kleinschreibung, die zehn arabischen Ziffern sowie einige Satz- und Steuerzeichen.
- 9 Inke Arns: Read_me, run_me, execute_me. Code als ausführbarer Text: Softwarekunst und ihr Fokus auf Programmcodes als performative Texte.
http://www.medienkunstnetz.de/themen/generative_tools/software_art/8/
- 10 <http://www.p0es1s.net/de/projects/jaromil.html>
- 11 <http://searchsongs.cyberfiction.ch/>
- 12 <http://searchsonata.netzliteratur.net/>
- 13 Benjamin Whooley, New Media-Worlds. London 1992, S.165.
- 14 Wreader = der Leser „Reader“, der auch Autor „Writer“ ist.
- 15 Uwe Wirth: Literatur im Internet. Oder. Wen kümmert's wer liest?, 1996
<http://www.netzliteratur.net/wirth/litim.htm>
- 16 Vergleiche dazu die instruktiven Ausführungen von Florian Hartling: Der digitale Autor. Autorschaft im Zeitalter des Internets, Bielefeld 2009, S. 321ff.
Dass es durchaus auch fruchtbare Schnittmengen zwischen Computerspiel und Netzliteratur gibt, zeigen Arbeiten von JODI und dem Schweizer Künstlerduo AND-OR (Beat Suter, René Bauer).
- 17 Leben im Cyberspace. William Gibson, der Erfinder des "Cyberspace" im Gespräch. In: c't - Magazin für Computertechnik, 15/2008, S. 204
- 18 Max Bense, Reinhard Döhl: Zur Lage, 1964
http://www.stuttgarter-schule.de/zur_lage.htm
- 19 Max Bense: Über natürliche und künstliche Poesie. A.a.O, S. 143ff.
Online: http://www.stuttgarter-schule.de/natuerliche_und_kuenstliche_poesie.html
- 20 Zur Lage, a.a.O.
- 21 Dieser schöne Vergleich stammt von Nicholas Carr: The Big Switch: Der große Wandel. Cloud Computing und die Vernetzung der Welt von Edison bis Google, Heidelberg 2009, S.254f.
- 22 <http://www.jodi.org/>
- 23 Von Interpreten wird immer betont, dass JODI so deutlich mache, dass hinter jedem Oberflächentext des Computers, ein Programmtext, ein Code steht. Übersehen wird dabei, dass JODI, bis auf den proprietären <blink> Tag, den Quelltext als erzeugenden Subtext eben gerade nicht thematisiert. Die Arbeit funktioniert ja nur, wenn die HTML-Tags im Quelltext weggelassen werden.
- 24 <http://auer.netzliteratur.net/kill/killpoem.htm>

25 Aya Karpinkas Performance "Lala" ist auf der E-Poetry Website leider nicht dokumentiert. Ein Mitschnitt findet sich hier: <http://www.youtube.com/watch?hl=de&v=q4p6Y0G4TkA>

26 <http://www.3durch3.de/>

27 Peter Weibel: Die Algorithmische Revolution. Zur Geschichte der interaktiven Kunst, 2004.
<http://www01.zkm.de/algorithmische-revolution/>