

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜNSTEN

FLORIAN CRAMER

ZUSAMMENFASSUNG. Computer und Internet unterscheiden sich von allen anderen sogenannten neuen Medien dadurch, daß sie auf einer Codierung basieren. Alle Information wird in ihnen als binärer Text prozessiert, und auch die Prozessierung selbst ist textuell gesteuert. Daß die spannendste Netzkunst der letzten Jahre sich besonders mit digitalen Codes befaßt hat und ästhetische Parallelen zu konkreter Poesie und anderer experimenteller Literatur zeigt, ist daher vielleicht kein Zufall. Dieser Vortrag unternimmt, mit vielen Beispielen, eine vergleichende Lektüre von papierner Dichtung und Computer-Codekunst, um beide sich wechselseitig erhellen zu lassen und der Poesie sich selbst ausführender Schriften auf die Spur zu kommen.

DICHTUNG UND PROGRAMMCODE

Beginnen möchte ich mit einem Gedicht des französischen Schriftstellers und Mathematikers François le Lionnais von 1972:

```
Table  
Begin: to make format,  
go down to comment  
while channel not false  
(if not true). End.
```

Dieser Text liest sich zunächst wie ein technizistisches Haiku oder ein Stück konkrete Poesie, in dem der Titel – die Tabelle – sich selbst beschreibt. Wer in den 1960er oder 1970er Jahren programmieren gelernt hat, erkennt darüber hinaus, daß das Gedicht nicht nur auf Englisch verfaßt ist, sondern auch in der damals verbreiteten Computer-Programmiersprache ALGOL (der modernere, Ihnen vielleicht bekanntere Programmiersprachen wie C, BASIC, Pascal und Java nachfolgten).

Date: 12.5.2002.

Dieses Detail ist von durchaus programmatischer Bedeutung. Le Lionnais war zusammen mit Raymond Queneau Begründer des „Oulipo“, einer Schriftstellergruppe, die es sich zur Aufgabe gemacht hatte, Literatur mit formalen und mathematischen Spielregeln zu verbinden. Neben Queneau und Le Lionnais gehörten bzw. gehören ihr unter anderem Georges Perec, Italo Calvino, Jacques Roubaud und der deutsche Lyriker Oskar Pastior an. 1962 forderte das erste Oulipo-Manifest, „bei Bedarf auf die guten Dienste von Datenverarbeitungsmaschinen zurückzugreifen“ und poetische „Vorstöße“ auf das „Gebiet besonderer Vokabularien (wie denen von Raben, Füchsen, Tümlern; die Programmiersprache von Computern - Algol - etc.)“ zu unternehmen.

Das Besondere an dem ALGOL-Vokabularium ist, daß Texte wie das oben zitierte Gedicht nicht nur doppelt, sondern zumindest theoretisch auch dreifach lesbar sind: Erstens als englischer Text, zweitens als Text in der Sprache Algol, und drittens, wenn seine Maschineninstruktionen ausgeführt werden, als der Output des ausgeführten Programms.

Genau um diese Irritationen und mehrfachen Lesbarkeiten zwischen Literatur und Computercodes soll es heute abend gehen.

Ein zweites Beispiel, das fast dreißig Jahre später (und übrigens in dieser Stadt) entstand:

```
$cd ug/models/soldier3
$origin 0 -6 24
$base base
$skin skin

$frame soldierc
$frame soldierd

/*

*/

void() army_fire;

void() army_stand1 =[ $soldierc,army_stand2 ]
{ai_stand();};
void() army_stand2 =[ $soldierc,army_stand3 ]
{ai_stand();};
void() army_stand3 =[ $soldierc,army_stand4 ]
{ai_stand();};
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
void() army_stand4 =[ $soldierc,army_stand5 ]
{ai_stand();};
void() army_stand5 =[ $soldierc,army_stand6 ]
{ai_stand();};
void() army_stand6 =[ $soldierc,army_stand7 ]
{ai_stand();};
void() army_stand7 =[ $soldierc,army_stand8 ]
{ai_stand();};
void() army_stand8 =[ $soldierc,army_stand1 ]
{ai_stand();};

void() army_walk1 =[ $soldierc,army_walk2 ] {
if (random() < 0.2)
sound (self, CHAN_VOICE, "soldier/idle.wav", 1, ATTN_IDLE);
ai_walk(1);};
void() army_walk2 =[ $soldierc,army_walk3 ]
{ai_walk(1);};
void() army_walk3 =[ $soldierc,army_walk4 ]
{ai_walk(1);};
void() army_walk4 =[ $soldierc,army_walk5 ]
{ai_walk(1);};
void() army_walk5 =[ $soldierc,army_walk6 ]
{ai_walk(2);};
void() army_walk6 =[ $soldierc,army_walk7 ]
{ai_walk(3);};
void() army_walk7 =[ $soldierc,army_walk8 ]
{ai_walk(4);};
void() army_walk8 =[ $soldierc,army_walk9 ]
{ai_walk(4);};
void() army_walk9 =[ $soldierc,army_walk10 ]
{ai_walk(2);};
void() army_walk10 =[ $soldierc,army_walk11 ]
{ai_walk(2);};
void() army_walk11 =[ $soldierc,army_walk12 ]
{ai_walk(2);};
void() army_walk12 =[ $soldierc,army_walk13 ]
{ai_walk(1);};
void() army_walk13 =[ $soldierc,army_walk14 ]
{ai_walk(0);};
void() army_walk14 =[ $soldierc,army_walk15 ]
{ai_walk(1);};
void() army_walk15 =[ $soldierc,army_walk16 ]
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
{ai_walk(1)};};
void() army_walk16 =[ $soldierc,army_walk17 ]
{ai_walk(1)};};
void() army_walk17 =[ $soldierc,army_walk18 ]
{ai_walk(3)};};
void() army_walk18 =[ $soldierc,army_walk19 ]
{ai_walk(3)};};
void() army_walk19 =[ $soldierc,army_walk20 ]
{ai_walk(3)};};
void() army_walk20 =[ $soldierc,army_walk21 ]
{ai_walk(3)};};
void() army_walk21 =[ $soldierc,army_walk22 ]
{ai_walk(2)};};
void() army_walk22 =[ $soldierc,army_walk23 ]
{ai_walk(1)};};
void() army_walk23 =[ $soldierc,army_walk24 ]
{ai_walk(1)};};
void() army_walk24 =[ $soldierc,army_walk1 ]
{ai_walk(1)};};

void() army_run1 =[ $soldierc,army_run2 ] {
if (random() < 0.2)
sound (self, CHAN_VOICE, "soldier/idle.wav", 1, ATTN_IDLE);
ai_run(11)};};
void() army_run2 =[ $soldierc,army_run3 ]
{ai_run(15)};};
void() army_run3 =[ $soldierc,army_run4 ]
{ai_run(10)};};
void() army_run4 =[ $soldierc,army_run5 ]
{ai_run(10)};};
void() army_run5 =[ $soldierc,army_run6 ]
{ai_run(8)};};
void() army_run6 =[ $soldierc,army_run7 ]
{ai_run(15)};};
void() army_run7 =[ $soldierc,army_run8 ]
{ai_run(10)};};
void() army_run8 =[ $soldierc,army_run1 ]
{ai_run(8)};};

void() army_atk1 =[ $soldierc,army_atk2 ] {ai_face()};};
void() army_atk2 =[ $soldierc,army_atk3 ] {ai_face()};};
void() army_atk3 =[ $soldierc,army_atk4 ] {ai_face()};};
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
void() army_atk4 =[ $soldierc,army_atk5 ] {ai_face();};
void() army_atk5 =[ $soldierc,army_atk6 ]
{ai_face();army_fire();

};
void() army_atk6 =[ $soldierc,army_atk7 ] {ai_face();};
void() army_atk7 =[ $soldierc,army_atk8 ]
{ai_face();SUB_CheckRefire (army_atk1);};
void() army_atk8 =[ $soldierc,army_atk9 ] {ai_face();};
void() army_atk9 =[ $soldierc,army_run1 ] {ai_face();};

void() army_pain1 =[ $soldierc,army_pain2 ] {};
void() army_pain2 =[ $soldierc,army_pain3 ] {};
void() army_pain3 =[ $soldierc,army_pain4 ] {};
void() army_pain4 =[ $soldierc,army_pain5 ] {};
void() army_pain5 =[ $soldierc,army_pain6 ] {};
void() army_pain6 =[ $soldierc,army_run1 ]
{ai_pain(1);};

void() army_painb1 =[ $soldierc,army_painb2 ] {};
void() army_painb2 =[ $soldierc,army_painb3 ]
{ai_painforward(13);};
void() army_painb3 =[ $soldierc,army_painb4 ]
{ai_painforward(9);};
void() army_painb4 =[ $soldierc,army_painb5 ] {};
void() army_painb5 =[ $soldierc,army_painb6 ] {};
void() army_painb6 =[ $soldierc,army_painb7 ] {};
void() army_painb7 =[ $soldierc,army_painb8 ] {};
void() army_painb8 =[ $soldierc,army_painb9 ] {};
void() army_painb9 =[ $soldierc,army_painb10] {};
void() army_painb10=[ $soldierc,army_painb11] {};
void() army_painb11=[ $soldierc,army_painb12] {};
void() army_painb12=[ $soldierc,army_painb13] {ai_pain(2);};
void() army_painb13=[ $soldierc,army_painb14] {};
void() army_painb14=[ $soldierc,army_run1 ] {};

void() army_painc1 =[ $soldierc,army_painc2 ] {};
void() army_painc2 =[ $soldierc,army_painc3 ]
{ai_pain(1);};
void() army_painc3 =[ $soldierc,army_painc4 ] {};
void() army_painc4 =[ $soldierc,army_painc5 ] {};
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
void() army_painc5 = [ $soldierc, army_painc6 ]
{ai_painforward(1);};
void() army_painc6 = [ $soldierc, army_painc7 ]
{ai_painforward(1);};
void() army_painc7 = [ $soldierc, army_painc8 ] {};
void() army_painc8 = [ $soldierc, army_painc9 ]
{ai_pain(1);};
void() army_painc9 = [ $soldierc, army_painc10]
{ai_painforward(4);};
void() army_painc10= [ $soldierc, army_painc11] {ai_painforward(3);};
void() army_painc11= [ $soldierc, army_painc12] {ai_painforward(6);};
void() army_painc12= [ $soldierc, army_painc13] {ai_painforward(8);};
void() army_painc13= [ $soldierc, army_run1] {};

void(entity attacker, float damage) army_pain =
{
local float r;

if (self.pain_finished > time)
return;

r = random();

if (r < 0.2)
{
self.pain_finished = time + 0.6;
army_pain1 ();
sound (self, CHAN_VOICE, "soldier/pain1.wav", 1, ATTN_NORM);
}
else if (r < 0.6)
{
self.pain_finished = time + 1.1;
army_painb1 ();
sound (self, CHAN_VOICE, "soldier/pain2.wav", 1, ATTN_NORM);
}
else
{
self.pain_finished = time + 1.1;
army_painc1 ();
sound (self, CHAN_VOICE, "soldier/pain2.wav", 1, ATTN_NORM);
}
};
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
void() army_fire =
{
local vector dir;
local entity en;

ai_face();

sound (self, CHAN_WEAPON, "soldier/sattck1.wav", 1, ATTN_NORM);

// dodging player
en = self.enemy;

dir = en.origin - en.velocity*0.2;
dir = normalize (dir - self.origin);

FireBullets (4, dir, '0.1 0.1 0');
};
```

```
void() army_die1 =[ $soldier,army_die2 ] {};
void() army_die2 =[ $soldier,army_die3 ] {};
void() army_die3 =[ $soldier,army_die4 ]
{self.solid = SOLID_NOT;self.ammo_shells = 5;DropBackpack();};
void() army_die4 =[ $soldier,army_die5 ] {};
void() army_die5 =[ $soldier,army_die6 ] {};
void() army_die6 =[ $soldier,army_die7 ] {};
void() army_die7 =[ $soldier,army_die8 ] {};
void() army_die8 =[ $soldier,army_die9 ] {};
void() army_die9 =[ $soldier,army_die10 ] {};
void() army_die10 =[ $soldier,army_die10 ] {};

void() army_cdie1 =[ $soldier,army_cdie2 ] {};
void() army_cdie2 =[ $soldier,army_cdie3 ]
{ai_back(5);};
void() army_cdie3 =[ $soldier,army_cdie4 ]
{self.solid = SOLID_NOT;self.ammo_shells = 5;DropBackpack();ai_back(4);};
void() army_cdie4 =[ $soldier,army_cdie5 ]
{ai_back(13);};
void() army_cdie5 =[ $soldier,army_cdie6 ]
```

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
{ai_back(3)};};
void() army_cdie6 =[ $soldier,army_cdie7 ]
{ai_back(4)};};
void() army_cdie7 =[ $soldier,army_cdie8 ] {};
void() army_cdie8 =[ $soldier,army_cdie9 ] {};
void() army_cdie9 =[ $soldier,army_cdie10 ] {};
void() army_cdie10 =[ $soldier,army_cdie11 ] {};
void() army_cdie11 =[ $soldier,army_cdie11 ] {};

void() army_die =
{
// check for gib
if (self.health < -35)
{
sound (self, CHAN_VOICE, "player/udeath.wav", 1, ATTN_NORM);
ThrowHead ("progs/h_guard.mdl", self.health);
ThrowGib ("progs/gib1.mdl", self.health);
ThrowGib ("progs/gib2.mdl", self.health);
ThrowGib ("progs/gib3.mdl", self.health);
return;
}

// regular death
sound (self, CHAN_VOICE, "soldier/death1.wav", 1, ATTN_NORM);
if (random() < 0.5)
army_diel ();
else
army_cdiel ();
};

/*UG monster_army (1 0 0) (-16 -16 -24) (16 16 40) Ambush
*/
void() monster_army =
{
if (deathmatch)
{
remove(self);
return;
}
precache_model ("progs/soldier.mdl");
```


WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
precache_model ("progs/h_guard.mdl");
precache_model ("progs/gib1.mdl");
precache_model ("progs/gib2.mdl");
precache_model ("progs/gib3.mdl");

precache_sound ("soldier/death1.wav");
precache_sound ("soldier/idle.wav");
precache_sound ("soldier/pain1.wav");
precache_sound ("soldier/pain2.wav");
precache_sound ("soldier/sattck1.wav");
precache_sound ("soldier/sight1.wav");

precache_sound ("player/udeath.wav"); // gib death

self.solid = SOLID_SLIDEBOX;
self.movetype = MOVETYPE_STEP;

setmodel (self, "progs/soldier.mdl");

setsize (self, '-16 -16 -24', '16 16 40');
self.health = 30;

self.th_stand = army_stand1;
self.th_walk = army_walk1;
self.th_run = army_run1;
self.th_missile = army_atk1;
self.th_pain = army_pain;
self.th_die = army_die;

walkmonster_start ();
};
```

Dieser Text wurde am 22. Oktober 2001 über den internationalen E-Mail-Verteiler "Rhizome" verschickt; seine Absender waren die in Barcelona beheimateten Netzkünstler jodi. Einen Monat vor den New Yorker Anschlägen konnte man diesen Text zum Beispiel als Kommentar auf die Mazedonien-Krise und ihre Diskussion auf Netzkultur-Mailinglisten lesen.

jodis Text ist ein ästhetisch beeindruckendes Stück konkrete Poesie. Die militärische Ordnung scheint dem Text in seine sprachliche und typographische Struktur eingeschrieben zu sein. Er wirkt wie eine codierte Inventarliste eines Waffenarsenals oder wie ein militärstrategischer Plan, oder

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

vielmehr wie eine Codesprache, die sich bizarr verselbständigt und autistische Logik entwickelt hat.

Zum Vergleich könnte man eine der klassischen „Konstellationen“ des bolivianischen-deutschen Begründers der konkreten Poesie, Eugen Gomringer, heranziehen, z.B. die erste der „3 variationen zu ‚kein fehler im system‘“ aus den späten 1960er Jahren (von der ich leider nur eine deutsche Fassung kenne):

kein fehler im system
kein efehler im system
kein ehfler im system
kein ehlfel im system
kein ehlefr im system
kein ehlerf im system
kein ehleri fm system
kein ehleri mf system
kein ehleri ms fystem
kein ehleri ms yfstem
kein ehleri ms ysfem
kein ehleri ms ystfem
kein ehleri ms ystefm
kein nehler ms systemf
fkei nehler im system
kfei nehler im system
kefi nehler im system
keif nehler im system
kein fehler im system

Auch hier handelt es sich gewissermaßen um einen programmierten Text. Der Fehler in Gestalt des „f“ schiebt sich Zeile für Zeile von links nach rechts, indem der Text mathematisch-kombinatorisch regelgerecht permutiert, d.h. in seiner Anordnung versetzt wird. Man dieses formale Verfahren, also diesen Algorithmus, in ein Computerprogramm schreiben, das ihn nicht nur auf den Satz „kein fehler im system“, sondern auch auf beliebige andere eingegebene Sätze anwendet.

Vielleicht stimmen Sie meiner Meinung zu, daß der Text von jodi unzweideutig besser ist als der von Gomringer: Sowohl als typographische Konstellation, als auch als Sprachmontage und Sprachreflexion ist er vielseitiger, geistreicher und ästhetisch interessanter. Tatsächlich handelt es sich bei ihm, genau wie bei dem kurzen Gedicht von François le Lionnais, um einen

Software-Quelltext, und zwar in der (heute weit verbreiteten) Programmiersprache „C“. In rein programmiertechnischer Hinsicht unterscheidet es sich sowohl von Le Lionnais', als auch von Gomringers System-Gedichten.

Gomringer unterschlägt den Quell- bzw. Rezeptcode seines Gedichts; er notiert allein die Ausführung des Algorithmus und zwingt den Leser damit, deren hypothetischen Quellcode deduktiv zu rekonstruieren. Dieses Verfahren hat den Nachteil, daß sich der niedergeschriebene Code durch 19 Zeilen mehr oder weniger redundanter Ausführungen einer einzigen simplen Idee quälen muß, während der Quellcode wie bei Le Lionnais und jodi nur die Idee notiert, um aus einer begrenzten und kompakten Notation eine fast unendliche Abundanz von möglichem Output zu generieren, Dichtung als Verdichtung im Medium des Programmcodes.

Im Gegensatz schließlich zu Le Lionnais' ALGOL-Programm, das sich, wenn man es auszuführen versucht, mit einer Fehlermeldung beendet, ist jodis C-Code tatsächlich lauffähig und benutzbar als Teil eines maschinenübersetzbaren Programmcodes:

Demonstration Quake II

Es handelt sich bei jodis konkreter Poesie nämlich um ein objet trouvé, ein Ready-Made aus einem Teil des Quellcodes des Ego-Shooter-Computerspiels „Quake“. Indem aber jodi mit ihrer Präsentation des Texts diese Herkunft und Funktion verschweigen, wird der Code in seinen ästhetischen und politischen Subtexten erst lesbar; jodi lesen den Quake-Autor John McCarmack gewissermaßen als Netzkünstler und schreiben seinen Code durch ihre Lektüre in die Geschichte konkreter Poesie und experimenteller Künste ein.

Denn eins haben jodi und andere Digitalkünstler – wie z.B. die ebenfalls in Barcelona lebenden und arbeitenden 0100101110101101.org und Joan Leandre – mit konkreter Poesie und sprachexperimenteller Dichtung gemeinsam: Kunst als ästhetisches Hacking, Auseinandernehmen und Sichtbarmachen von Strukturen. Darüber hinaus gibt es auch medienhistorische bzw. semiotische Gründe für die offenkundigen Parallelen zwischen experimenteller Netzkunst und experimenteller Dichtkunst. Das Internet ist, nach dem Morsefunk, das erste moderne Medium – oder vielmehr die erste moderne Informationstechnologie – die auf einem Code basiert. Wenn man „Text“ allgemein definiert als eine Kette einzelner (diskreter) Zeichen, die aus einem endlichen Repertoire (einem Alphabet) entnommen sind, so sind auch 01-Codes Texte und alle digitalen Technologien textuelle Technologien. Sogenannte Softwarearchitekturen, das Internet eingeschlossen, sind selbst nichts als Text; Text, der in Computersprachen geschrieben ist und als

digitaler Schriftcode übermittelt, transformiert und ausgeführt wird. Das Internet ist ein Schriftgebilde aus komplex gewobenen Codes von Betriebssystemen, Hilfsprogrammen, Programmiersprachen und Netzwerkprotokollen, von denen „Hypertext“ und „World Wide Web“ nur eine äußerste und völlig arbiträre Repräsentationsschicht sind. Denn das Neue ist nicht, daß man Computer als vernetzte Bildschirmlesegeräte oder nur als simple Expansionen anderer Medien nutzen kann, sondern daß digitale Texte bzw. Codes auch Programme sein können, die die Übertragungen und Prozessierungen anderer Codes steuern. Deshalb kann zum Beispiel eine E-Mail-Nachricht eben nicht nur eine Textbotschaft sein, sondern auch ein Virus, der den Computer des Empfängers zum Absturz bringt.

Störungen, Inkompatibilitäten und Fehlcodierungen Teil ihrer Poetik und Ästhetik. Indem sie den Computer vor allem als selbstbezüglichen Generator kontingenter Zeichen versteht, als System von Kontrolle und Absturz, ist ihr Verständnis digitaler Zeichen skeptischer und auch politischer als das einer Literatur, die mit Hypertext-Verweisen und digitaler Vernetzung immer noch eine ästhetisch-gesellschaftliche Utopie verbindet.

An dem „Quake“-Quellcode und seiner Entwendung durch jodi zeigt sich, daß die wirklich gute Netzliteratur meistens diejenige ist, die sich selbst nicht so nennt, oder, anders gesehen, wie schwierig es ist, aus dem Codegebilde des Internets ein spezielles literarisches Feld zu destillieren und von Nicht-Literatur zu unterscheiden. Während sogenannte multimediale Netzliteratur digitale Technologie kurioserweise als Auflösung von Text begreift und damit eine alte Geschichte des Selbsthasses von Literatur und Literaturwissenschaft fortschreibt, hat die bildende Netzkunst die umgekehrte Entwicklung eingeschlagen, ihr Material auf das Wesentliche – nämlich Codes – zu reduzieren und Abstürze, Inkompatibilitäten und Fehlcodierungen nicht auszublenden, sondern in ihre Poetik und Ästhetik zu integrieren. Der Computer wird somit nicht naiv als Wunschmaschine gesehen, sondern humorvoll und kritisch als Generator kontingenter Zeichen und System von Kontrolle und Absturz.

Wenn ich also Arbeiten, die sich selbst *Netzkunst* nennen, als Literatur und Dichtung wahrnehme, dann nicht, um ihnen diese Bezeichnung überzustülpen, sondern um meine persönliche Leseweise zu akzentuieren.

TECHNOPÄGNIE UND ASCII ART

Die Differenz von Quellcode und Ausgabecode kennzeichnet nicht nur Computerprogramme, sondern auch triviale Text-, Graphik- und Audiodateien, so zum Beispiel auch HTML-codierte Webseiten. Diese doppelte Lesbarkeit thematisiert eine andere Arbeit von jodi, „Location“. Klickt man

durch die Webseiten, so sieht man Zeichen, die an dadaistische Montagen oder Stör-codes erinnern:

Jodi, Location, eine Seite in der Browser-Darstellung

Die Arbeit erschließt sich erst, wenn man sich im Web-Browser den Quellcode der Seiten anzeigen läßt:

Jodi, Location, Quellcode

Was im Fenster des Webbrowser wie Störzeichen aussieht, ist im Quellcode ein traditionelles ASCII- Kitschbild eines Teddybärs. Die Webbrowser-Software selbst codiert, ohne weitere Programmierung, die Teddybären in ein nichtmimetisches Schriftbild um, da sie die Zeilen des Typogramms neu umbricht und sein Raster damit zerstört. Die ästhetische Erwartung an digitale Zeichen, in der formatierten Darstellung „lesbar“ und im Quellcode „unlesbar“ zu sein, wird hier auf den Kopf gestellt. Indem die Arbeit mit ihren sichtbaren und verborgenen Zeichen spielt, erfordert sie Leser-Forscher, die den technischen Unterschied von Anzeige und Quellcode kennen und beide Zeichenebenen zu entziffern wissen.

Wie bei dem vermeintlichen konkrete Poesie-Kriegsgedicht sind auch bei „Location“ Insider-Referenzen in den Code eingeschrieben. Jodi verweisen hier auf die „ASCII Art“ und ihre Geschichte. Seit 1963 ist ASCII, der „American Standard Code for Information Interchange“, der kleinste gemeinsame Nenner aller Computer- Systemschriften: ein Alphabet von 128 Buchstaben, Zahlen und Interpunktionszeichen, das dem Zeichenvorrat einer amerikanischen Schreibmaschine entspricht. ASCII-Text ist Computertext ohne jegliche typographische Auszeichnungen, ohne Schriftformatierungen also, und bildet nach wie vor den Basiscode für E-Mail sowie das Zeichenrepertoire aller Programmiersprachen und der meisten Netzwerkprotokolle. Bevor die graphischen Benutzeroberflächen des Apple Macintosh und von Microsoft Windows Graphiken und formatierten Text einführen und bevor das World Wide Web diese Neuerung aufs Internet übertrug, spielte sich alle Netzkommunikation im ASCII-Code ab. Wo dies als Mangel empfunden wurde, entstanden analog zu mechanisch getippten Schreibmaschinen-Graphiken elektronische ASCII-Typogramme, die Fotografien und Zeichnungen durch Buchstaben und Zahlen nachbildeten. Mittlerweile kann spezielle Software Bild- automatisch in Buchstabenraster übersetzen:

Demonstration: display, asciiview;

Demonstration: mplayer, mplayer -vo aa

Somit erfand die frühe, gegenständliche ASCII Art aus der Not der technischen Restriktion ein Textgenre neu, das seit der griechischen Antike bekannt ist, nämlich das der Technopägnie bzw. des Figurengedichts. Zu den ältesten überlieferten Figurengedichten gehören die „Flügel des Eros“ des Simias von Rhodos, die um das Jahr 300 vor Christus entstanden sind:

[Original, Adler/Ernst]

Übersetzung:

Siehe mich, der die breitbrüstige Erde beherrscht und den
Sohn des Akmon anderswohin setzte

Zittre nicht, wenn obschon ich klein bin, meine schattigen
Wangen schwer sind von wolligem Haar

Denn ich wurde damals geboren, als die Notwendigkeit
herrschte.

Der Erde unseligen Satzungen folgte alles

Kriechende und alles, was sich regt
im Äther.

Des Chaos,

nicht der Kypris

und des Ares schnellfüßiger Sohn wurde ich genannt.

Denn keineswegs herrschte ich mit Gewalt, sondern mit
sanftmütiger Überredung;

es fügten sich mir die Erde, die Winkel des Meeres und der
eherne Himmel;

diesen entwand ich das uralte Szepter und gab den Göttern
Gesetze.

Das Gedicht imitiert die Flügelform allein durch sein absteigendes und wieder ansteigendes Versmaß. Wie Jeremy Adler und Ulrich Ernst anmerken, wird dieser geflügelte Eros nicht als Liebesgott beschrieben, sondern im Kontext orphischer Kulte (– und wie im Renaissance-Neuplatonismus von Marsilio Ficino –) als Schöpfer der Welt. Vermutlich war das Gedicht die Inschrift eines flügelförmigen Altars; auf Altarformen und ihren Beschriftung werden auch die anderen klassischen Formen der Technopägnie, das Beil und das Ei abgeleitet. Im 17. Jahrhundert greift der englische metaphysical poet George Herbert die Flügel des Simias auf und christianisiert sie zu seinem Gedicht „Easter Wings“. Auch in der Moderne reißt die gegenständlich-mimetische Tradition der Figurendichtung nur scheinbar ab. Mallarmés „Coup de dés“ endet mit der typographischen Nachbildung eines Sternbilds und der von der konkreten Poesie später aufgegriffenen Zeile „UNE CONSTELLATION“. Marinettis futuristisch „befreite

Wörter“ dienten zur typographisch-phonetischen Imitation von Maschinen und Kriegsschauplätzen.

Raoul Hausmann, kp'erioum

Auch in dadaistischen Figurengedichten wie Raoul Hausmanns „kp'erioum“ ist die experimentelle Typographie vor allem phonetische Stimmpartitur. Die „Konstellationen“ Gomringers und der konkreten Poesie lösen sich erstmals konsequent vom Mimetismus, doch schreibt sich in der bloßen Bezeichnung „Konstellation“ immer noch der Mimetismus von Mallarmés Sternenbild fort.

Im Gegensatz zu dieser Tradition sind Buchstaben und Zahlen in der digitalen ASCII Art nur visuelle Raster ohne schriftsprachliche Semantisierung. Mit der Netzkunst von jodi, Vuk Cosic und aus dem Umfeld der Mailingliste „7-11“ wandelt sich die ehemals naiv-gegenständliche ASCII Art zu einer Kunst, die ASCII-Codes als Spielmaterial für sich entdeckt.

www.7-11.org

Einerseits macht diese ASCII Art Codierungen von Daten sichtbar, indem sie sie im einfachsten Textformat ausschreibt, andererseits irritiert sie damit ihre Betrachter. Kommerzielle PC-Betriebssysteme ASCII-Text betrachten nur noch als Rauschen oder Funktionsstörung. Auf Windows-PCs markiert er Betriebsmeldungen der Hardware und – als „Blue Screen of Death“ – den Systemabsturz. In der experimentellen ASCII Art dreht sich diese ästhetische Normierung um, und Fehlercodes, Protokollzeichen, Rauschen und Systemmeldungen treten in den Vordergrund. In einem technischen System, das wie das Internet selbst von Code gesteuert wird, können solche Zeichen tatsächlich als technische Störung aufgefaßt werden. So wurde der jodi.org-Webserver vor zwei Jahren abgeschaltet, weil der Provider die Website für einen Computervirus hielt. Für das poetische Programm der Absturzcode-Poesie genügt der ästhetische Schein des Systemfehlers, solange er den Leser irritiert. Das Betriebssystem – und der technische Angriffspunkt – der Codepoesie ist daher weniger die Maschine, als die Imagination des Lesers.

Jodis „Location“ kartographiert den Übergang von der naiv-gegenständlich zur ungegenständlich-disruptiven ASCII Art, indem es sich die doppelte Lesbarkeit digitaler Daten als Quellcode und dessen formatierter Output zunutzemacht: Die alte ASCII Art wird hier wörtlich zum Quellcode der neuen ASCII Art.

REKURSION

Definiert man Computerviren als einen ausführbaren Code, der sich potentiell unendlich oft selbst repliziert und fortschreibt, so könnte man ihre Geschichte nicht erst in den frühen 1980er Jahren beginnen lassen, wie es etwa Robert M. Slade's „History of Computer Viruses“ tut. Als Prototypen könnte man etwa frühneuzeitliche Anagramm- und Wortpermutationsgedichte wie Quirinus „XLI. Libes-kuß“ von 1671 anführen, ein Sonnet über den „Wechsel menschlicher Sachen“, das in seinen vertauschbaren Wörtern Makro- und Mikrokosmos permutieren läßt und über das sein Autor in einem Nachwort schreibt: „nn aber einer Beliben hegte / aus dem ersten vierverse in funffzig Wörtern einen Wechselsatz zu volführen / so konten alle Menschen / wenn si solchen darzustellen gleich trachteten / dises nicht ausmachen / weil es so vilmahl tausendmahl tausend verwechselt werden können / daß auch nur di Meer-sandkörner / welche dise Zahl beschlüsse / unser virgetheilte Kugelrund nicht begriffe.“ (Paraphrase in modernem Deutsch: „Wenn aber einer die ersten fünfzig Wörter aus der ersten Gedichtstrophe permutieren wollte, so würde dies auch allen Menschen zugleich nicht gelingen, weil die Wörter soviele Millionen mal permutiert werden können, daß die Zahl sogar der der Sandkörner auf der Erde übertreffen würde.“).

Auf Nacht / Dunst / Schlacht / Frost / Wind / See / Hitz /
 Süd / Ost / West / Nord / Sonn / Feur und *Plagen* /
 Folgt Tag / Glantz / Blutt / Schnee / Still / Land / Blitz /
 Wärmd / Hitz / Lust / Kält / Licht / Brand und *Noth*:
 Auf Leid / Pein / Schmach / Angst / Krig / Ach / Kreutz /
 Streit / Hohn / Schmertz / Qual / Tükk / Schimpf / als
Spott /
 Wil Freud / Zir / Ehr / Trost / Sig / Rath / Nutz / Frid / Lohn
 / Schertz / Ruh / Glück / Glimpf / stets *tagen*.
 Der Mond / Glunst / Rauch / Gems / Fisch / Gold / Perl /
 Baum / Flamm / Storch / Frosch / Lamm / Ochs / und
Magen
 Libt Schein / Stroh / Dampf / Berg / Flutt / Glutt / Schaum /
 Frucht / Asch / Dach / Teich / Feld / Wiß / und *Brod*:
 Der Schütz / Mensch / Fleiß / Müh / Kunst / Spil / Schiff /
 Mund / Printz / Rach / Sorg / Geitz / Treu / und *GOtt* /
 Suchts Zil / Schlaß / Preiß / Lob / Gunst / Zank / Port / Kuß
 / Thron / Mord / Sarg / Geld / Hold / *Danksagen*
 Was Gutt / stark / schwer / recht / lang / groß / Weiß / eins /
 ja / Lufft / Feur / hoch / weit *genennt* /

Pflegt Böß / schwach / leicht / krum / breit / klein / schwarz
/ drei / Nein / Erd / Flutt / tiff / nah / *zumeiden* /
Auch Mutt / lib / klug / Witz / Geist / Seel / Freund / Lust /
Zir / Ruhm / Frid / Schertz / Lob muß *scheiden* /
Wo Furcht / Haß / Trug / Wein / Fleisch / Leib / Feind / Weh
/ Schmach / Angst / Streit / Schmerz / Hohn *schon rennt*
Alles wechselt ; alles libet ; alles scheint was zu hassen:
Wer nur disem nach wird=denken / muß di Menschen Weiß-
heit fassen.

Kuhlmann projizierte 1674 eine „ars magna librum scribendi“, die auf kombinatorischem Wege sämtliche vorhandenen und sämtliche künftigen Bücher erzeugen sollte, eine Utopie, die parodistisch im Lagado-Kapitel von Swifts „Gulliver’s Travels“ wiederkehrt und als poetisch-epistemologische Spekulation in Borges’ „Bibliotheca di Babel“.

Doch während die „Bibliotheca“ trotz aller Komplexität endlich ist, ist das „Frametale“ des amerikanischen Schriftstellers John Barth aus dem 1968 erschienenen Prosaband „Lost in the Funhouse“ tatsächlich eine unendliche Erzählung, die noch dazu mit zehn Wörtern Quellcode auskommt. Es besteht aus einem Satz auf einem Möbiusband:

ONCE UPON A TIME THERE WAS A STORY THAT BE-
GAN

Ähnlich wie im Lied vom Mops in der Küche, das Beckett in „Warten auf Godot“ zitiert, wird hier eine endlose Geschichte erzählt. Die kaum übersetzbare Eleganz des englischen Texts liegt darin, daß sich in ihm zwei Sätze ineinander verschieben, „ONCE UPON A TIME THERE WAS A STORY THAT BEGAN [as follows:]“ und „THERE WAS A STORY THAT BEGAN ONCE UPON A TIME“. So zirkuliert diese Geschichte nicht einfach, sondern verschachtelt in sich selbst, indem sie bei jedem neuen Durchlauf der Schleife eine neue erzählerische Klammer öffnet.

In Mathematik und Computerprogrammierung ist eine Funktion, der sich selbst aufruft, indem sie ihren Output wieder in den Inputkanal speist, eine Rekursion. Das „Frametale“ ist somit eine erzählerische Rekursion par excellence. Rekursiv ist ferner auch die Erzählung des chinesischen Philosophen Chuang Tzu, der träumt, ein Schmetterling zu sein und nach dem Aufwachen nicht mehr weiß, ob er Chuang Tzu ist oder ein Schmetterling, der träumt, Chuang Tzu zu sein. Musikalische Rekursionen finden sich, wie Douglas R. Hofstadter nachweist, in Bachs Fugen, paradoxe Aussagen wie „Dieser Satz ist eine Lüge“ sind logische Rekursionen. Robert Altmans

Film „The Player“ endet rekursiv, als dem Filmproduzenten, dessen Geschichte erzählt wurde, von einem Drehbuchautor vorgeschlagen wird, seine Geschichte zu verfilmen. In Programmierer-Terminologie ausgedrückt, führen sich diese Erzählungen selbst aus, indem sie sich endlos selbst aufrufen.

Man könnte daher das „Frametale“ nicht nur als Rahmenerzählung von „Lost in the Funhouse“ lesen, sondern auch als Meta-Rekursion, als Meta-Erzählung nämlich aller rekursiven Erzählungen. Darüber hinaus ist es das seltene Beispiel einer poetischen Rekursion, in der der Code selbst – d.h. die Buchstaben – sich unendlich verschleift, während die Filmrolle des „Players“ und die Borges’ Erzählung Anfang und Ende besitzen.

Als Programmiermethode und intellektuelles Spiel waren Rekursionen besonders unter Systemprogrammierern am Artificial Intelligence Lab des MIT beliebt, die in den frühen 1970er Jahren eines der ersten Netzwerkbetriebssysteme schrieben und für sich das Wort „Hacker“ erfanden. MIT-Hacker übertrugen Rekursion von Programmiersprachen auf die Umgangssprache, indem sie rekursive Akronyme für ihre Programme erfanden. Das bekannteste von ihnen ist „GNU“, das Präfix der freien Softwareprogramme der Free Software Foundation, die einen Hauptbestandteil des heute als „Linux“ bekannten Betriebssystems bilden; „GNU“ steht für „GNU’s not Unix“, das, weil sich bei jeder Auflösung des Akronyms wieder „GNU“ an den Anfang setzt, wie folgt expandiert:

```

G                N   U
G                N   U's Not Unix
G      N   U's   Not Unix Not Unix
GNU's Not Unix  Not Unix Not Unix
...

```

Im Gegensatz zu den Rekursionen von Beckett und John Barth verschleift sich im poetischen Spiel der rekursiven Akronyme nicht eine Erzählung, sondern das bloße Wort. Es wird zu einem Prozeß, der sich selbst steuert und übersteuert. Die Selbstausführbarkeit von digitalem Programmcode überträgt sich hier in die Umgangssprache, das Wort wird zu einem sich selbst replizierenden Virus. Darin gleichen rekursive Akronyme zwar älteren sprachkombinatorischen Dichtungsformen wie buchstabenwechselnden Anagrammen oder wortwechselnden Proteusversen wie Kuhlmanns „XLI. Libes-Kuß“. Doch während Permutationen bei aller Abundanz immer endlich sind, wuchern rekursive Strukturen nicht nur scheinbar, sondern auch faktisch unendlich und katastrophisch.

MEZ, _VIRO.LOGIC CONDITION][ING][1.1_

Einen gedanklichen Schritt über programmierte Dichtung und Programmiersprachen-Poesie hinaus geht Netzdichtung, die ihre Sprache an zwar an Computer-Programmiersprachen, rekursiven Strukturen und ASCII-Protokollcodes geschult hat, aber keine strikte Befehlssyntax verwendet, sondern imaginäre Codes kreiert. Die australische Netzkünstlerin mez (Mary Anne Breeze) schreibt in einer selbsterfundenen Kunstsprache „mezangelle“, die stärker noch als andere Netzkunst aus dem ehemaligen Umfeld der Mailingliste 7-11 zu sprachexperimenteller Lyrik hin tendiert.

Den letzten Teil meines Vortrags möchte ich mez' Text „_Viro.Logic Condition][ing][1.1_“ widmen. Da der Text nicht einfach ist, beschränke ich mich hier darauf, seine erste, kürzere Hälfte Schritt für Schritt zu lesen:

Viro.Logic Condition][ing][1.1

[b:g:in]

::Art.hro][botic][scopic N.][in][ten][dos][tions::

1.[b.ranch outwards][seething
jam-jar curs][ed][ored
drenching s][creening][ounds]

::Neol][o.jism][ithic Rever][b][s.al][l][s::

2.[drink sever][al][ed
c u in he][l][avan
a c][yclops][hair b:cumming sane]

::Gig:a][h!][:cycling::

3.[alert & c.rash.ing
chrysa][s][li][ding!][s//via
code syrup & brooding symbols]

Beginnen wir mit dem Titel:

Viro.Logic Condition][ing][1.1

[b:g:in]

Wie in zum Teil in E-Mail und in Diskussions-Newsgroups üblich, werden hier einfache Unterstriche („_“) an Zeilenanfang und -ende als ASCII-Zeichenersatz für hervorgehobenen Text verwendet. Mez' wichtigstes Stilmittel sind eckige Klammern, deren Gebrauch sich an Klammerausdrücke in Booleschen Suchausdrücken oder sogenannten regulären Suchausdrücken in Programmiersprachen und Suchmaschinen anlehnt. Demnach kann der Titel als „Virologic Condition“ („virologischer Zustand“) gelesen werden oder als „Logic Condition“ („logischer Zustand“), „Logic Conditioning“ („logische Konditionierung“) oder „Virologic Conditioning“ („virologische Konditionierung“).

Wie ein Computerprogramm, ist dieser Text mit einer Versionsnummer versehen, und so, wie in vielen Programmiersprachen, wird der eigentliche Code zwischen einer „begin“- und einer „end“-Zeile umklammert. Durch die der internationalen phonetischen Schrift entlehnten Doppelpunkte wird „begin“ jedoch zu (englisch pronunciert) „bee-gee-in“ gedehnt, einem phonetischen Bastard von „begin“ („Anfang“) und „being“ („sein“).

Die ersten beiden Zeilen lassen offen, wer oder was konditioniert oder einem Virus ausgeliefert wird: Ein menschlicher Körper oder ein technisches System? Diese Frage durchzieht sich als roter Faden durch den Text.

Ähnlich wie in der Programmiersprache Perl, folgen nun drei „Pakete“ bzw. Objekte, die Unterprogramme enthalten, und voneinander durch mit „::“ gekennzeichnete Anfangszeilen abgetrennt sind. Die erste lautet:

```
::Art.hro][botic][scopic N.][in][ten][dos][tions::
```

Wieder werden mit den eckigen Klammern Schachtelwörter kreiert bzw. „portmanteu words“ nach dem Vorbild von Lewis Carroll und James Joyce. Das erste Wort wird dadurch vierfach lesbar als „Arthroscopic“ („arthroskopisch“), „Art robotic“ („kunstrobotisch“), „Arthrobotic“ („arthro[se]botisch“) und „horoscopic“ („horoskopisch“). Das zweite Wort expandiert zu „Nintendos“, „intentions“ und (dem Namen des alten Microsoft-/IBM-Betriebssystems) DOS.

Die Sprache verschmilzt Maschinen mit menschlicher Anatomie, so daß aus Maschine und menschlichem Bewußtsein ein grotesker Hybridkörper entsteht:

```
1.[b.ranch outwards||seething
```

Wie auf der Kommandozeile des Unix-Betriebssystems und in den Programmiersprachen C und Perl steht der doppelte Längsstrich („||“) für eine logische „oder“-Bedingung. Diese Zeile kann zweifach gelesen werden:

- (1) als Verben: „branch outwards, seething“ („nach außen zerfließen und aufgescheucht sein“); eine Beschreibung dessen, was der virologische Hybridkörper tut.
- (2) als Substantive: „ranch outwards, see-thing“ („eine Ranch draußen, ein See-Ding“): eine Beschreibung der Außenlandschaft.

jam-jar curs] [ed] [ored

Hier ist im Gegensatz zur vorigen Zeile die Doppelbedeutung nicht syntaktisch, sondern semantisch codiert:

- (1) „jam-jar cursed“: ein Marmeladentopf wird verflucht, als Innenbeschreibung gegenüber der Außenbeschreibung „ranch outwards“
- (2) „jam-jar cursored“: mit einem Maus- oder Tastaturcursor, der wie ein Marmeladentopf aussieht. – Nun also wechselt die Sprache zum Computerbildschirm.

„jam-jar“ liest sich außerdem als Anspielung auf „Jar-Jar“, einem ersten photorealistisch computergenerierten Charaktere eines Spielfilms, der in „Star Wars Episode No.1“ (2000) auftritt und Vorwürfe rassistischer Stereotypisierung sowie den Haß von der Star Wars-Fangemeinde in Hacker-Netzforen wie Slashdot.org auf sich zog. Neben „jam-jar cursed“ lauert hier also auch „jar-jar cursed“ zwischen den Zeilen.

drenching s] [creening] [ounds]

„drenching screening“/„drenching sounds“ („absaufende Vorführung“, „absaufende Töne“). Betrachtet man das gesamte vierzeilige „Objekt“, so ist es schwierig zu sagen, wer oder was das agierende Subjekt des Texts und wer oder was das Objekt: Der arthritische Roboter, der virus-infiziert und mit Marmeladentopf-Cursor, absaufendem Bildschirm und Tönen nach außen zerfließt, oder aber die „athroscopischen Intentionen“ des schreibenden Subjekts in seiner häuslichen Umgebung?

Zum nächsten Programmobjekt:

::Neol] [o.jism] [ithic Rever] [b] [s.al] [l] [s::

Wieder werden Wörter durch eckige Klammern geschachtelt: „Neolithic“ („neolithisch“), „Neologism“ („Neologismus“), „jism“ („Sperma“), sowie „Reverb“, „Reverbs“ („hallen“), „Reversal“ („Umkehrung“), „Reversals“ („Umkehrungen“), „all“ („alle“).

Zwei mögliche, fast autonome Lesarten: Das Subjekt fühlt, als ob es in die Steinzeit zurückversetzt wäre oder fühlt die Steinzeit nachhallen; oder aber es gibt einen „neologism reversal“, eine Umkehrung eines Neologismus.

Neben dem Leitmotiv der Vermischung von menschlichem Körper und Technik ist als zweites Leitmotiv bereits Ansteckung und Krankheit zu erkennen. Der Text kann aufgrund seiner Mehrfachcodierungen konsequent als ein privater Krankenbericht gelesen werden und ebenso konsequent als Bericht von viral infizierter Technologie. Vor diesem Hintergrund ist „neologism reversal“ eine sehr geradlinige Anspielung: Das biologische Virus wird, qua Neologismus, zum Computervirus und durch die Umkehrung des Neologismus vom Computervirus wieder zum biologischen Virus. Das sprechende Subjekt beobachtet also seine eigene Krankheit, deutet sie wie eine Ansteckung mit einem Computervirus und begreift, daß es den Neologismus umkehrt, indem sie sich das biologische anhand des digitalen Virus erklärt.

2.[drink sever][al][ed

„drink several“/„drink severed“ („trink einige“, „das Getränk abgeschnitten“): Der Krankenbericht schreibt sich paradoxisch fort. Entweder wird Flüssigkeit eingenommen, um die Infektion zu bekämpfen, oder das Getränk ist, wie der „jam-jar cursed“, das „drenching screening“ und der „jism“, eine vergiftete Quelle und ein Multiplikator der Ansteckung.

c u in he][1][avan

„See you in hell“/„see you in heaven“. Natürlich ist das der offensichtliche zusammengezogene Gegensatz im Text, aber es stecken noch zwei Internet-spezifische Anspielungen in der Zeile. „c u“ bezieht sich auf die Videokonferenz-Software „cu/see me“, „see you in hell“ imitiert die Rhetorik von Computerviren und von technisch primitiv arbeitenden „Script Kiddie“-Hackern/Crackern.

a c][yclops][hair b:cumming sane]

Diese Zeile expandiert zu „a cyclops becoming sane“/„a chair becoming sane“/„hair becoming sane“ („ein genesender Zyklop“, „ein genesender Stuhl“, „genesendes Haar“), dabei ist „to come“ zweideutig „to cum“ („kommen“/einen Orgasmus haben) eingeschrieben, so daß „b:cumming“ für „becoming“ („werden“) und „be cumming“ („gerade kommen“) steht.

Da die vorherige Zeilen auf Web-Cameras („c u“) anspielt, erscheint „cyclops“ („Zyklop“) als Metapher für den Computer als einäugiges Wesen. Wieder ist es mehrdeutig, wer angesteckt wurde und wer gesund wird. Denn selbst in der Genesung lauert, durch den eingeschriebenen Sex-Slang, die potentielle Übertragung und Infektion.

Zum nächsten Programmobjekt:

WENN SCHRIFT SICH SELBST AUSFÜHRT: ÜBER CODES UND IHRE REFLEXION IN LITERATUR UND DIGITALEN KÜ

```
>  
> ::Gig:a][h!][:cycling::
```

„Gigahigh cycling“/„Gigahertz cycling“/„Gig cycling“/„Gigah! cycling“: Diese wegen der Mehrdeutigkeit von „cycling“ schwer ins Deutsche übersetzbare Lesarten beschreiben, je nach dem, die Gigahertz-Taktgeschwindigkeit eines Computers, Fahrradfahren in den Bergen und ein extremes Gefühl des Auf und Ab. In „Gigah_i“ steckt eine Anspielung auf die einschlägig bekannte Codesprache der umstrittenen Netzkünstler antiorp/Netochka Nezvanova, in welcher der Buchstabe „i“ konsequent durch Ausrufezeichen ersetzt wird. Da dies die einzige Stelle in mez’ mit solch einer Buchstabenersetzung ist, liest sie sich als verstecktes „Hi“ an mitleidende Kollegen.

```
3.[alert & c.rash.ing
```

„alert & crashing“ / „rash“ („aufgeschreckt & zusammenbrechend“ / „rücksichtslos“)

Dieses Objekt bzw. dieser Strophenabsatz beschreibt, wie auch in der nächsten Zeile deutlich wird, den Organismus bzw. die Maschine im destabilisierten Zustand:

```
chrysa][s][li][ding!][s//via
```

„chrysalis“/„sliding_i“, // via („Larve“, „gleitend“, „via“)

Mit der Larve wird eine weitere biologische Metapher eingeführt und per mezangelle zu „chrysaliding“ verschachtelt. Der Organismus/die Maschine läuft zugleich auf Hochtouren außer Kontrolle („Gigacycling“, „rash“, „alter & crashing“) und zieht sich in den atavistischen, regressiven Status einer Larve zurück. Die Larve gleitet via:

```
code syrup & brooding symbols]
```

Wieder mischen sich organische Biologie und Computerprogrammierung. Der „Syrup“ ist zugleich Medizin des rekonvaleszenten schreibenden Subjekt, und, als Code, das Reparaturmittel für die abgestürzte Maschine. „Codesyrup und Brutsymbole“ erscheinen als Metaphern der Computersteuerung und -programmierung, die durch Code und Symbole einen organisch empfundenen System kontrollieren. Das erste Absatz-Objekt des Texts beschreibt somit den hybriden Organismus, das zweite, das ihn in einen mehrdeutig codierten räumlichen Kontext rückt, die Quellen der Infektion und der Heilung, und das dritte Absatz-Objekt die innere Verfassung des angesteckten Hybridorganismus.

Mit dem „Code-Syrup“ und den „Brutsymbolen“ in der letzten Zeile reflektiert sich schließlich auch der mezangelle-Text selbst in seinem Versuch, den irrationalen Status quo durch Schreiben zu rationalisieren.

Im der nachfolgenden Prosa-mezangelle, die ich hier aus Zeitgründen auslasse, ist noch expliziter vom E-Mail-Virus Sir.Cam die Rede, und von Physis und Logik als Kanälen trojanischer Einwanderung. Zum Schluß imitiert der Text die Typographie und die Sprache einer elektronischen „manpage“, also der Referenz für ein Unix-Kommandozeilenprogramm. Die „Virologic Condition“ ist also eine imaginäre Software, und mez' Text liefert ihre technische Dokumentation. Allerdings ist diese Dokumentation schon selbst wieder Software, Text nämlich, der qua seiner geklammerten Schachtelungen wie permutative Gedichte aus einem knappen Quellcode algorithmisch expandiert.

ANALYTIKER VERSUS SYNTHETIKER

Vergleicht man jodi und mez mit zu antiker und frühneuzeitlicher Figurdichtung, Permutationslyrik des Barock und der Moderne, mit Oulipo-Dichtung und konkreter Poesie, so fällt vor allem ein Unterschied auf: Im Gegensatz selbst zur experimentellen Dichtung der 1970er Jahre verfahren die ASCII-Montagen und Programmcode-Werke der Netzkunst nicht synthetisch, sondern analytisch. Sie sind keine säuberlichen Reinraum-Konstruktionen aus dem Labor, sondern eklektische Ready-Mades und Umcodierungen von vorgefundenem Code. – Das gilt noch viel mehr für die Arbeit von 0100101110101101.org und Joan Leandre, auf die ich nicht näher eingegangen bin. – Auch wenn mir Pauschaletikettierungen wie „Postmoderne“ versus „high modernism“ generell widerstreben, so scheinen sie mir hier nicht verkehrt zu sein. Der Bruch zwischen beiden Ansätzen, der sich in Quellcode und Formatcode von jodis „Location“ widerfindet, wurde erst durch Personal Computer und das Internet möglich. War Programmcode in den 1960er und 1970er Jahren, aber auch noch in der Frühzeit der Heimcomputer, etwas, das man immer eigenhändig schreiben und konstruieren mußte, so wurde er mit Aufkommen von kommerzieller Benutzersoftware erst scheinbar unsichtbar gemacht, um darauf im Internet sich um so massiver aufzudrängen. Spätestens seit der Massennutzung des Internets gibt es nicht mehr zuwenig, sondern zuviel Computercode als Spielmaterial, was aus der Sicht des Beobachters und offenbar auch aus der Sicht von Künstlern nur zu begrüßen ist.

©Dieser Text unterliegt der Open Publication License Version 1.0 <http://opencontent.org/openpub/> und darf gemäß ihrer Bestimmungen frei kopiert und weiterverwendet werden.