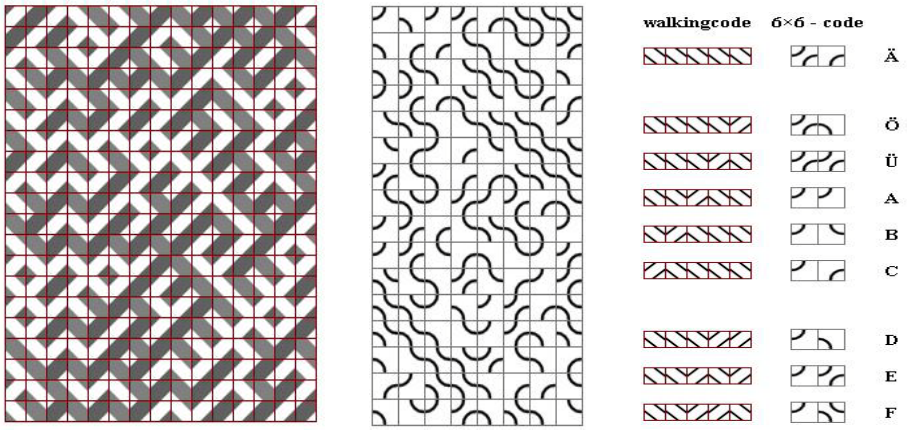
5. Program Code poetry

Auf seinen Internetseiten mit dem Titel “untexte” weist **Manfred Arens** auf die unendliche Vielfalt der Kodierungsmöglichkeiten hin, die natürlichen Sprachen stellen nur einen kleinen Ausschnitt aus dem breiten Spektrum dar. Er überträgt Gedichte in verschiedene Codes wie musikalische Notenschrift, farbige Stäbchen, Rechtecke, Bodenfliesen und andere graphische Muster.



Die abgebildeten Grafiken sind verschlüsselte Gedichte. Wenn man den Code kennt, kann man das linke Bild als die Zeichenfolge “Wer will erfaren der Welt Wesen, der thuo disen Reimen lesen” entschlüsseln. Das mittlere Muster repräsentiert den Satz “Der untext verhält sich zur leeren Seite, wie der Text zum Untext”. Es ist in einem Code, den der französische Mönch Sébastien Truchet im 18. Jahrhundert für Fliesen entwickelte, nachempfunden (vgl. LABOR INTUS).
Die von Arens verwendeten Codes könnten ohne weiteres zur Steuerung von Computern verwendet werden. Sie weisen auf die doppelte Codierung von digitalen Texten hin. Unter der Oberfläche der natürlichen Sprache verbirgt sich eine Kunstsprache, die Programmiersprache, die ihrerseits für den Rechner wieder in binären Code umgewandelt wird. Es ist daher wenig überraschend, wenn Cyber-Autoren versuchen, Programmiersprachen poetische Effekte zu entlocken oder zumindest mit Elementen von Programmiersprachen zu spielen. In diesem Fall spricht man von *Program Code Poetry* oder *Code Work*. Diese Dichtungsgattung wird definiert als “literature which uses, addresses and incorporates code [...] as a special type of language in itself, or as an intrinsic part of the new surface language or ‘interface text’, as I call it, of writing in networked and programmable media.” Auch diese Experimente weisen auf den Unterschied zwischen Oberflächentext und der Tiefenstruktur, die ihn generiert, hin, zwischen menschlicher Lektüre und Maschinentext. Die Programmiersprache, in der der Source Code verfasst ist, nimmt eine vermittelnde Rolle zwischen der für uns lesbaren Oberfläche und der Maschine ein. Der Netzkünstler Jaromil bemerkt dazu:

> Source code means a formulation of ‘instructions’ expressed in a language understandable to a computer and linked in accordance with logical and conditional patterns which, once interpreted and executed, gives rise to a result. [...] Every language is defined by a

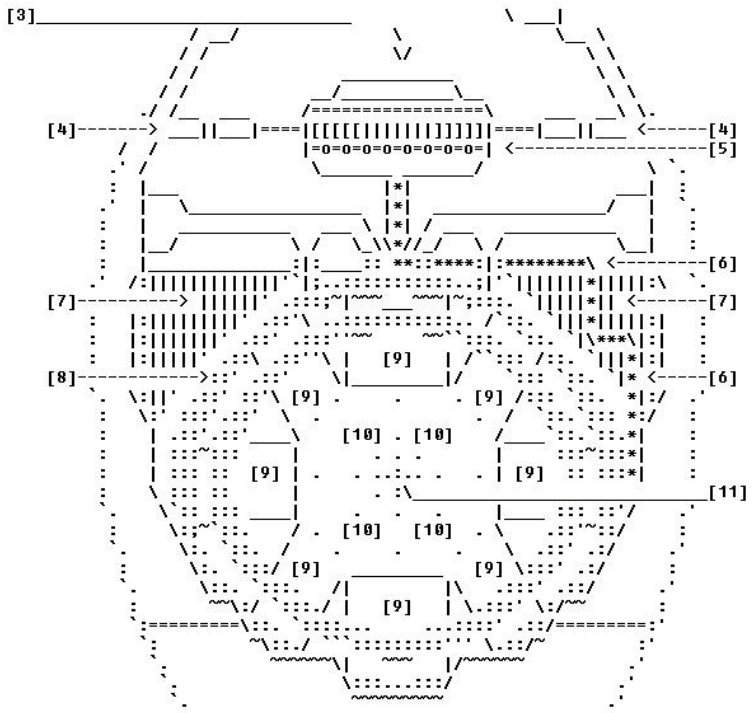
> grammar which, in turn, is interpreted by a compiler who ‘metabolizes’ its semantic content (instructions) and so produces a ‘bytecode’ which the computer can execute.

Und Lawrence Lessig erläutert die unterschiedliche Transparenz und den unterschiedlichen Komplexitätsgrad der beiden Code-Schichten unter der Oberfläche:

> Source code is the code that programmers write. It is close to a natural language, but not quite a natural language. A program is written in source code, but to be run it must be converted into a language the machine can read. Some source code is converted on the fly [...]. But most source code - or the most powerful source code - is ‘compiled’ before it is run. The compiler converts the source code into either assembly code (which mavens can read) or object code (which only geniuses and machines can read). Object code is machine-readable. It is an undifferentiated string of 0s and 1s that instructs the machine about the tasks it is to perform. Programmers do not directly write object code, even if some are able to decipher it; programmers write source code. Object code speaks to the computer; source code speaks to humans and to computers (compilers); assembly code speaks to mavens and computers. (103)

Der *Source Code* repräsentiert die unsichtbare und für den durchschnittlichen user unzugängliche Welt der Maschine, in der Bedeutung durch strikt logische Prozesse generiert wird. Wenn ein Programm, ‘läuft’, bewirkt es etwas, und zwar im materiellen Sinn. Daher hat man *Source code* mit illokutionären Sprechakten verglichen, die zur gleichen Zeit etwas aussagen und bewirken bzw. performieren. Diese Performativität der Sprache wird betont, wenn der *Source Code* für künstlerische Zwecke an die Oberfläche geholt wird.
Jodi ist ein Akronym von **Joan Heemskerk** und **Dirk Paesmans**. Dieses holländisch-belgische Künstlerduo betreibt verschiedene Netzkunstprojekte. Hier sind ihre Installationen, in denen sie den ASCII-Code zur Herstellung von Texten oder Bildern in der Tradition der Konkreten Poesie verwenden, von besonderem Interesse. ASCII (American Standard Code for Information Interchange) umfasst die 128 Buchstaben, Zahlen, Interpunktions- und Zusatzzeichen einer amerikanischen Schreibmaschinentastatur. Dieser Zeichensatz bildet nach wie vor den Basiscode für den e-mail-Verkehr und Programmiersprachen. Viele von Jodis Arbeiten bestehen aus a-semantischen Zeichenketten, die aussehen wie der Bildschirm eines infolge Systemfehlers abgestürzten Computers. Jodi durchbrechen die Oberfläche des Informationssystems und simulieren ein Code-Chaos. Wenn man den *Source code* dieser Installationen betrachtet, verwandelt sich das scheinbare Chaos in Ordnung - oder zumindest in ein regelmäßiges Muster. Die ‘normalen’ Verhältnisse - lesbarer Oberflächentext, kryptischer Quellcode - sind hier umgekehrt. Die Zeichen sind dieselben geblieben, aber ihre durch den Webbrowser veränderte und dadurch ‘unsinnige’ Anordnung ergibt jetzt Sinn bzw. ein Bild wie im folgenden Beispiel. Zuerst wird im Webbrowser ein “Zeichensalat” sichtbar.

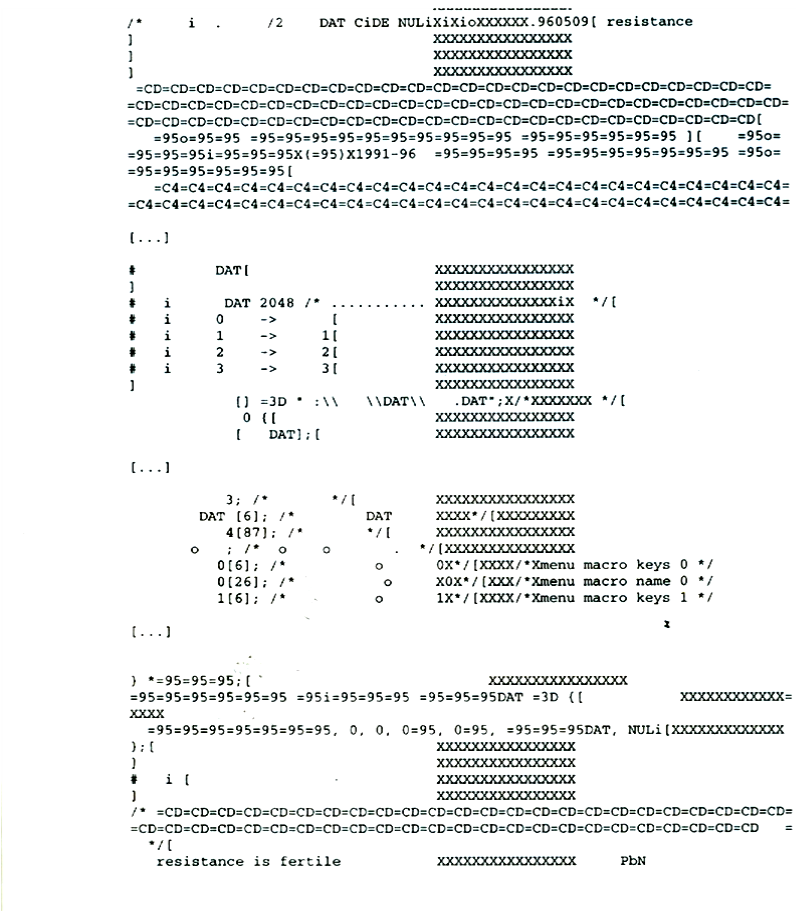
Erst der Quellcode zeigt das zugrundeliegende Muster, man könnte auch sagen: das Kunstwerk.



ASCII-Kunst geht zurück auf Versuche, mit der Tastatur zu zeichnen, und sei es so primitive *icons* wie den Smiley :-). Aufwändiger sind Projekte wie Deep ASCII, ein von der Gruppe ASCII hergestellter Text-Film, der das Vorbild, den Pornofilm *Deep Throat*, digitalisiert und zu einem bewegten Zeichenraster umwandelt. Es ist kaum etwas zu erkennen, außer vagen Umrissen und beweglichen Formen. Entfernt erinnert das Ergebnis an Bilder von verschlüsselten Pay-TV-Sendern. Auch hier geht es um Fragen der Verschlüsselung und Dekodierung bzw. der Lesbarkeit.

Im digitalen Medium sind alle Darstellungen (auch Bilder, Musik …) unter der Oberfläche textuell kodiert. Kodierung und Darstellung treten zum Unterschied von analogen Medien denkbar weit auseinander. Die normalerweise verdrängten Quellcodes werden in digitaler

Kunst zuweilen an die Oberfläche geholt, einschließlich der Fehlermeldungen und Störfaktoren, deren subversives Potential entsprechend betont wird.
Der Kanadier **Ted Warnell** etwa verfremdet in seinem Text DAT CiDE NULi: eine Raumschiff-Enterprise-Dialogzeile (“you will be assimilated - resistance is futile” zu “resistance is fertile”).



```
                                                  ------------------
/*     i   .     /2     DAT CiDE NULiXiXioXXXXXX.960509[ resistance
]                                                 XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
 =CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=
=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=
=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD[
    =95o=95=95 =95=95=95=95=95=95=95=95=95=95 =95=95=95=95=95=95 ][     =95o=
=95=95=95i=95=95=95X(=95)X1991-96  =95=95=95=95 =95=95=95=95=95=95=95=95 =95o=
=95=95=95=95=95=95[
    =C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=
=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=C4=

[...]

#          DAT[                                   XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
#    i      DAT 2048 /* ..........                XXXXXXXXXXXXXXiX  */[
#    i      0    ->      [                         XXXXXXXXXXXXXXXX
#    i      1    ->     1[                         XXXXXXXXXXXXXXXX
#    i      2    ->     2[                         XXXXXXXXXXXXXXXX
#    i      3    ->     3[                         XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
             [] =3D * :\\    \\DAT\\    .DAT*;X/*XXXXXXX */[
              0 ([                                XXXXXXXXXXXXXXXX
              [   DAT];[                          XXXXXXXXXXXXXXXX

[...]

            3; /*        */[                     XXXXXXXXXXXXXXXX
         DAT [6]; /*         DAT                 XXXX*/[XXXXXXXXX
            4[87]; /*        */[                 XXXXXXXXXXXXXXXX
       o    ; /*  o     o          .  */[XXXXXXXXXXXXXXX
            0[6]; /*           o      0X*/[XXXX/*Xmenu macro keys 0 */
            0[26]; /*           o     X0X*/[XXX/*Xmenu macro name 0 */
            1[6]; /*            o     1X*/[XXXX/*Xmenu macro keys 1 */

[...]

} *=95=95=95;[                                  XXXXXXXXXXXXXXXX
=95=95=95=95=95=95 =95i=95=95=95 =95=95=95DAT =3D ([           XXXXXXXXXXX=
XXXX
  =95=95=95=95=95=95=95=95, 0, 0, 0=95, 0=95, =95=95=95DAT, NULi[XXXXXXXXXXXXX
};[                                               XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
#    i [                                          XXXXXXXXXXXXXXXX
]                                                 XXXXXXXXXXXXXXXX
/* =CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=
=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD=CD    =
  */[
    resistance is fertile           XXXXXXXXXXXXXXXX     PbN
```

Ausschließlich Maschinencode benützt auch **Giselle Beiguelman** in ihren *Code Movies*. Sie entnimmt hexa, ASCII, und binäre Codes aus Bildern im jpg-Format, speichert sie als Text ab und bearbeitet sie mit Hilfe des Programms *Flash player* nach. Das Projekt weist auf die unsichtbaren Übersetzungsvorgänge im digitalen Medium hin und wirft die Frage nach der Rolle des *Source code* bei der Bedeutungskonstruktion auf (vgl. Code Movie1).
Ein Gedicht in Programmiersprache kann als Oberflächentext oder als Kette von Befehlen gelesen werden, zumindest wenn der Leser bzw. die Leserin Programmierkenntnisse mitbringt. In diesem Fall kann man sich auch die ungefähren Auswirkungen der Befehle ausmalen, ähnlich wie man als versierter Leser eines Drehbuchs oder einer musikalischen Partitur eine Vorstellung von der möglichen Realisierung entwickelt. Der poetische Gebrauch von Programmcode wirft die Frage nach den Unterschieden zwischen natürlicher und Maschinensprache auf, er verweist aber auch auf das prinzipielle Wirkungspotential jeder sprachlichen Äußerung. Außerdem kann man nie hundertprozentig sicher sein, dass das ‘Gedicht’ nicht doch irgendetwas im Computer bewirkt, auch Viren, Würmer und Co. treten schließlich als nützliche Besucher getarnt auf. Programmieren wird zuweilen auch als Kunst betrachtet und mit dem literarischen Schreiben verglichen. Sogar die Produktion und die Verbreitung von

Viren wird zumindest in Hackerkreisen als eine Art Kunst angesehen. Jedenfalls ist es nicht allzu überraschend, dass Programmcode für künstlerische Zwecke verwendet wird.
Lyrik in Programmiersprachen existiert vereinzelt seit den sechziger Jahren. Auch in diesem Bereich hat das OuLiPo Pionierarbeit geleistet. Das OuLiPo-Manifest von 1962 fordert unter anderem den Rückgriff auf Programmiersprachen. 1972 verfasste François Le Lionnais Gedichte in Algol-Code, die sich zum Beispiel folgendermaßen lasen:

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

Ein Haiku von **Larry Wall**, dem Erfinder der Programmiersprache PERL (Practical Extraction and Report Language), die hier auch verwendet wird, lautet:

```
Print STDOUT q
Just another Perl hacker
Unless $pring
```

Liest man den Text laut (q = queue, $pring = dollar spring), erfüllt das Gedicht die Anforderungen an ein Haiku, nämlich Zeilen von fünf, sieben und fünf Silben aufzuweisen und eine Jahreszeit zu erwähnen. PERL ist nicht so weit entfernt von natürlicher Sprache, aber es verfügt nur über ein sehr beschränktes Vokabular von ca. 250 Wörtern, mit denen man als Dichter das Auslangen finden muss. Extreme Verdichtung ist das Markenzeichen des folgenden anonymen Liebesgedichts.

```
sub merge{
    my $enses;
```

Ein interessanter Aspekt der PERL-Gedichte ist der Umstand, dass sie tatsächlich als Programme laufen können und dann einen anderen (poetischen?) Text, den maschinenlesbaren Text, produzieren. Man hat sogar vier Kategorien von PERL-Gedichten unterschieden: 1) nicht-funktionale, 2) als Programme ausführbare, 3) *keyword poems*, die ein PERL-Befehlswort als Ausgangspunkt benützen, und 4) humoristische *wordsalad poems*.
Mehr Raum für Kreativität bieten Texte, die natürliche und Computersprache mischen. Es entstehen dadurch hybride Texte oder Texte, die den strikt logischen Aufbau von Code als Muster für poetische Sprache verwenden. Die natürliche Sprache wird sozusagen durch *Program code* kontaminiert. Das Augenmerk wird auf die Tatsache gelenkt, dass *Program code* in den digitalen Unterhaltungsmedien allgegenwärtig und insbesondere bei der Textproduktion maßgeblich beteiligt ist. Gleichzeitig fordern hybride Texte die menschliche Wahrnehmung heraus und werfen die Frage nach der Grenze auf, bis zu der Texte lesbar bleiben. Im Hintergrund stehen Theoretiker einer posthumanen Kultur, die vorhersagen, dass unsere Kultur von einem Zeichensystem usurpiert wird, das nur auf unterschiedlichen

elektrischen Ladezuständen beruht und dazu geeignet ist, Menschen und Maschinen zu vernetzen und damit letztlich einander anzugleichen. Die spielerische *Program Code poetry* schwankt zwischen dem Akzeptieren des Unausweichlichen und dem Protest gegen die Mechanisierung der Kommunikation. Es kann als Demonstration des Siegeszugs der digitalen Technologie verstanden werden, wenn die poetische Sprache von Maschinenzeichen durchsetzt wird, wenn sich *dots* und *file extensions* einmischen wie *cyborgs* unter 'echte' Menschen, ein * eine beliebig zu füllende Leerstelle andeutet oder .tmp eine vorübergehende und später durch eine andere ersetzte Funktion bezeichnet.

Einfache Formen der *Code poetry* verwenden Zahlen und einige Zeichen und Symbole aus Computersprachen als Ersatz für Buchstaben oder Wörter und verzerren und verfremden so die natürliche Sprache. Der Code und damit das für Laien äußerst geheimnisvolle Geschehen in den Rechnern wird zumindest teilweise sichtbar. Zutage treten dabei die in allen Sprachen vorhandenen autoritären Kontrollmechanismen. Es überrascht nicht, dass das hybride *Codework* sich oft mit Fragen von Identität und Gender sowie der Technologie und ihrem Einfluss auf das menschliche Verhalten beschäftigt und globalisierungskritische Töne anschlägt.

Mez (Pseudonym für **Mary-Anne Breeze**), eine Hauptvertreterin dieser Textsorte, legt großen Wert auf den Dialog mit den Leserinnen und Lesern. Ihre Art zu schreiben geht auf kreativ ausgestaltete und verfremdete e-mails zurück, momentan führt sie die spielerisch mit Elementen von Programmiersprache versetzten Texte in das *social network*-Forum *LiveJournal* ein. Sie beruft sich dabei auf Techniken des so genannten *Social Tesseracting*: Darunter versteht man synthetische Interaktionen, die Verbindung von virtueller und realer sozialer Kommunikation, wie sie z. B. bei Spielern in einem multi player game, die einander auch im Leben treffen, oder bei realen face-to-face-Antworten auf e-mails auftritt; auch Phänomene wie die Einberufung von Demonstrationen per sms und Menschen, die ihr Leben in einer Art lifestream in Twitter dokumentieren, fallen unter diesen Begriff.

Mez nennt das von ihr verwendete poetische Idiom "mezangelle" (mit Anspielung auf 'angel', 'mangle', 'angle', 'elle'). Sie beschreibt es als ein polysemisches Sprach- bzw. Codesystem, das "attempts to expand traditional text parameters through layered/alternative meanings embedded in languages and the codes that create them." Zur Hybridisierung der Texte dienen insbesondere Klammern und *dots* zusammen mit Allgemeingut gewordenen Abkürzungen wie 2 für to, 4 für for, 10 für ten, z. B. in con10t, txt für text, lol für laugh out loud oder lots of love, cu für see you und ähnliches. Wörter werden manipuliert, auffällig sind insbesondere die häufigen Portmanteau-Wörter (Schachtelwörter), das Spiel mit Allusionen und Homophonen, das Einbauen von Symbolen sowie von "tree structures, wildcard refs, boolcanisms, unix shell commands, bare html conventions, ascii][esque][tracks [as well as the repeated allusions 2 hyperlinks and html code via bracketing & directory slashings] - all act 2 illustrate the x.pansion of software potentialities of co:d][iscours][e". Eine andere Selbstbeschreibung lautet: "2 _mezangelle_ means to take words/wordstrings/sentences and alter them <in> such a way as to enhance meaning beyond the predicted or the expected. It is similar to making "plain" text hypertextual via the arrangement, dissection, and splicing of code scripting practices in2 english." Diese 'Augmentierung' der natürlichen Sprache mit Elementen der Programmiersprachen geht auf den Jargon von Hackern zurück, die statt Buchstaben Zahlen in ihre Botschaften einsetzten, um sie zu verschlüsseln. Statt von Social Tesseracting spricht

man hier besser von Augmented Reality, der durch virtuelle Elemente angereicherten geophysikalischen Realität.
Mez erzeugt kleine Hypertexte, man könnte auch von permutativen Textgeneratoren sprechen, die als Spielmaterial zur Verfügung gestellt werden. In einem Interview verweigert Mez explizit jede Sinngarantie: “i *never* advocate a complete understanding or outright interpretation of my output”; und sie empfiehlt “2 play + to employ curiosity when mezangelle-navigating”. Ein Beispiel vom 29.01.2010 lautet:

> ***_modern:body::forces_ [or: How I Learned 2 Luv Those _Body_(c)Hunks_]***
> Part 1: _Body_(c)Hunking_
>
> "...four-dimensional [4D]at[a]mospheric Body_(c)Hunks depict the standard modern_body [redNet-merge-blueNet-merge-greenSoc] M-print. Current (c)Hunk construction proceeds as follows:
>
> 1. StRipp[l]ing of Intimacy:
>
> This type of 4D Biorippling ensures the tracking, trickling and eventual flattening of preemptive desire nodes. Biorappelling then removes all normative [hor]Monal traces and leave the entity [com]Punction-free.
>
> 2. Introduction of Intricacy:
>
> Stripping intimacy quotients removes the problem of -X[un]tended Affective [de]Formation. The introduction of [re]Placeable_Intricacy assures the conversion of [phero]Monal trajectories to [vec]Tor[n]Space_[di]Stancing in a sequence of required attitudes defined as functions of S[t]ex[t]ing.
>
> 3. Soft_S[t]ex[t]ing:
>
> System dynamics are then separated into the BioDepth[charged] + Cog[knit]Dependent. The CogKnitters are independently controlled and employ 4[D] performative [de]functions...."

Wie meist in Mez’ Texten geht es auch hier um die Identität im cyberspace, um Körperlichkeit und um das Schreiben im neuen Medium. Der Körper ist *data-spheric*, gehört der vierten Dimension (des virtuellen Raums?) an. Die Intimität geht in *social networks* verloren, auch Hormone und Gewissen werden nicht mehr gebraucht. An die Stelle der Intimität treten die Komplexität, das Schreiben und die damit verbundene sexuelle Identität im Cyberspace. Alle diese Botschaften müssen durch Erprobung der verschiedenen möglichen Lesarten aus dem Text extrapoliert werden.
Das auffälligste Merkmal von “mezangelle”, das Aufspalten von Wörtern, lädt zur Bildung neuer Silbenkombinationen und Bedeutungen ein und erinnert damit entfernt an *Finnegans’ Wake*. Typisch für weite Bereiche der von Code kontaminierten Poesie ist der folgende selbst-

reflexive Text, der - wie viele Mez-Kreationen - in der Form eines e-mails oder E-Forum-Diskussionsbeitrags verfasst ist. Die ersten “mezangelle”-Texte sollen tatsächlich Bearbeitungen realer e-mails gewesen sein. Hier ist es eine unter dem Pseudonym “hu][bris wo][man” firmierernde Schreiberin, die sich zum Thema “text|code text|code ::exe.][elo]cution text|code text|code” äußert.

.the
.randomness
.x.ists in the formulation

.coded][with][in their own

[hash d.finely ch[l]o[o]pped]
braille|small|emailness]

.semi-random but][ted [
.potentiali-T x.panse

% smaille editor = (

[omitted: lines of program code signs, composed of Xs, commas and various forms of brackets]

_
sparks of lost hu][bris][man §cent][+re.sieved][

Poetische Botschaften haben immer etwas Zufälliges, Beliebiges, die Netzkommunikation neigt zum Verkürzen und Verballhornen von Wörtern und bringt ein Idiom hervor, das für Uneingeweihte ähnlich fremd anmutet wie z. B. die Blindenschrift. Code kann ausgeführt werden, das Subjekt, das sich als Hybris erweist, ist verloren, es bleiben nur schwache Funken davon übrig, die ‘resieved’ werden können. Neben den *dots* verwendet Mez hier hauptsächlich eckige Klammern, die in vielen Programmiersprachen Mehrdeutigkeit signalisieren. Eine ähnliche e-mail-Botschaft von “Phonet][r]ix” zum Thema “<][w][Rit][e][ua.LIS][P]tic>” handelt von “<SCRATCH Language=Juven][v]ilescrypt>”, was sich eventuell auf das Kratzen (Scratchen) von Schallplatten durch DJs bezieht und auf eine ramponierte Jugendsprache verweist, die der “Art Of][E][Go][a]-Blasting” dient. Ein weiteres Beispiel von “mezangelle” dreht sich um das Rollenspiel in *chat rooms* und den so genannten *multi user dungeons*.

:: Fantazee Genderator::> Assig.n[ation]inge Ov Charact.wh[m]orez 2
[w]Re[ck]quired Fiction.all.lie.sd Para.m[edical] Statuz

:: Vari.able[bodie]z::> Prince Cessspit N Princess Pit N Cin.der.ella[fitzgeraldingz] N Rap[t]punzelle N Gr.etal]

:: Will B Mild[h]er than me[aslez] but damaging to the fe[male]tus during
the first try[mester].

[5 Micah Dolls awai.ting AC.TIF.[f]ASHION]

Der spielerische *chat* wirkt als Fantasie-Gen(d)erator, der fiktionale Charaktere hervorbringt, die, weil man sie nur zeitweise benützt, mit Huren verglichen werden. Hier nehmen ein Prinz Cessspit, eine Prinzessin Pit, Cinderella (Fitzgerald), Rapunzel und Gretel teil. Der dritte Absatz scheint der Konversation entnommen zu sein, er könnte aber auch eine Selbstreflexion eines der Avatare sein.
Der Einbruch des *Program code* in die natürliche Sprache steht im Mittelpunkt von Mez' .::...:the DataH Inpho[mill]ennium:.:..:, das von **Christy Sheffield Sanford** in ein Projekt mit dem Titel "'My' millennium", in dem verschiedene Künstler etwas zu diesem Thema beitrugen, aufgenommen wurde. Travestien von Bibelstellen und Gebeten, illustriert durch religiöse Motive, die durch Ketten von Binärcode verfremdet werden, läuten hier mit viel Ironie das neue Millennium ein, das eine schöne neue Welt der Information mit sich bringt, in der das $-Zeichen herrscht.

Then i sawe a C0mmandE @ the d0llah pr0mpte:
holdinge in itz teXt the keys of M: I: L: E: N: & U. 1t seiZed sum
0v the lettaHs: & K0dEd them up in2 a Th0usand N-crypti0nz. The
C0mmandE sent the N-crypti0nz staticscreaminge t0 the Binne.
The zer0Hs & w0nnes then flick.erred with joy, N the
DataHK0de was b0rnE with the re:maining lettaHz, the I: the
E: N 0therz B-sides, the P: & H: & O: all linked N clicking
straighte in2 the Inph0ennium...

Die Genesis wird hier mit dem Programmieren verglichen. Die Bibel, aber auch der Anfang des neuen Millenniums, des elektronischen Zeitalters, das durch die Götter der Berechnung (computation) eingeläutet wird, erscheinen in äußerst ironischem Licht. Die unterschiedlichen Bedeutungen von *command* (Befehl in der Computersprache, mit Anklang an *commandment* = biblisches Gebot) wird hier angesprochen, ebenso wie das Dollarzeichen, das in vielen Programmiersprachen verwendet wird und in *Program Code poetry* auf die Okkupation des Internet durch die Geschäftswelt aufmerksam macht. Computertechnik beruht auf "encryptions", auf Übersetzung in eine Programmiersprache, die ebenso geheimnisvoll ist wie manche Stellen der Bibel. Die übersetzten Texte werden an eine "bin(ne)" gesandt - laut *Oxford English Dictionary* "a receptacle" und insbesondere "each of a series of ranges of numerical value into which data are sorted in statistical analysis" - und dort der binären Umkodierung unterzogen, also in Ketten von Nullen und Einsern umgewandelt.
Ein anderer Teil des Textes ist mit "Pr0phetiCk-t0ck" überschrieben:

N we shell l00k 2 the screen, N on those screens we shell d1nd the line----_---__
and the d0t:: . . , and the d0t shell speax. N the d0t will teXt:

$take up yr keys and type
$N click yr modemz :0N:
$N scroll the pages thru
$N delete @b0minations
$N n0.s N linkz that breakE the koDE
$N thenne merge the stringZ
$N u shell B phree.

Das Dollarzeichen, das bereits in der ersten Lexia vorkam, unterstreicht hier den ‘prophetischen’ Charakter von Quellcode. Die Hochstellung des Wortes “screen” verdeutlicht seine herausragende, fast kultische Stellung. Alles in allem zeichnet sich das Gedicht durch die Verarbeitung und Verfremdung von aus dem Computerjargon entlehnten Wörtern aus. Der Text, der zugleich auch einen Hypertext darstellt, schließt mit einer Hymne auf das Internet:

010010100Pray.err@//Praising the DataH- - -
DataH, let uz pray
2 the l0gin & the password:
the 1RC and the !CQ:
the 1Fs & Thennes:
the zer0Hs n the w0nnes:
Let us swhymn thru the netw0rks
N web into the fabr1c of yr f0nts:
Bin[ary]d typ0h and tract
in2 rhiz0mes 0v yr k0de.

Dazu noch ein Kommentar von Mary-Anne Breeze: “This period [.] of strategically driven hype[rtime] intersects perfectee with the cult of the Information Age - both in scope & paranoia. We seem 2 cult.u.rally invest similar amounts ov consumptive eNerGY in2 PMT [P/re/sent/post Millennium Tension] & hy[pah!]steria, as 2wardz our tandem ack!knopwledgement of technology as sa[tanic]viour and potenti.all religion re:placement.”
Ein außergewöhnlich umfangreicher Text, der *Program code* verarbeitet, ist Lexia to Perplexia von **Talan Memmott**, einem Medienkünstler und Designer. Das Werk besteht aus zehn verlinkten Webseiten, auf denen sich zum Teil programmierter Text befindet. Die Installation ist instabil und präsentiert sich ‘nervös’, Texteinheiten verschwinden oder verändern sich oft schnell. Auch das ist ein Teil der Reflexion des Verhältnisses von user, Bildschirm und Netzwerken. Memmott vertritt die These, dass Mensch und Technik gleichberechtigt sind und einander im Verlauf der Geschichte gegenseitig hervorbringen. Körper werden mit elektronischem Material gleichgesetzt, beide sind daher starken Wandlungen unterworfen. Die Begegnung von user und Rechner führt zu der im Titel genannten perplexia (Verwirrung). In einem Interview erläutert Memmott, dass er darunter “a confusion of ontological, literary and technical application” versteht. Die Texte bewegen sich an der Grenze der Lesbarkeit, was aber nur folgerichtig ist, wenn Mensch und Maschine, Fleisch und

elektronisches Material miteinander kommunizieren. Immer wieder kehrt das Begriffspaar head - body. Gleich im *opening screen* des ersten von vier Kapiteln findet sich die Zeile:

<HEAD>[FACE]<BODY>, <BODY> FACE </BODY>

Hier ist sowohl von Körperteilen als auch - wenn man den Text als HTML-Code liest - von Textteilen die Rede; übersetzt bedeuten die zwei Phrasen: ‘Face ist Teil von Head, aber nicht von Body’ bzw. ‘Face ist Teil des Körpers’. Head und Body sind aber auch wesentliche Bestandteile eines Quelltextes. Der Quelltext zur Lexia Anonymus [N] lautet zum Beispiel:

```
<HTML>
<HEAD>
<TITLE>Anonymus.[N]</TITLE>

<script language“JavaScript“>
<!—
function loaded() {
if (parent.location.href.index0f(“FRAME.html” != -1)
parent.update(); // only called when inside preloader frameset
}
// →
</script>
<style>
body {background-color:#000000;font-
family:courier,arial,helvetica; font-size 10pt}
A{color:#336033; text-decoration:none;}
A:hover{color:#FFFFFF}
A:visited:{color:#c0c0c0;}

…
```

Alles, was zwischen <HEAD> (= opening tag) und </HEAD> (= closing tag, hier nicht mehr enthalten) steht, gehört sozusagen zum Head. Derselben Beziehung widmet sich auch ein Song, der wiederum natürliche und Programmiersprache vermengt. Ein Satz darin lautet:

```
I will conceal (to you, my dear) {
(this body.skin) {
```

Das kann als syntaktisch etwas eigentümlicher, aber noch verständlicher englischer Satz durchgehen. Vermutlich will das sprechende Ich seine Haut vor jemand anderem (my dear) verbergen. Es folgen mehrere Umformungen des Satzes, in denen das Ich sagt, dass es verwundbar (vulnerable) ist, sich schließlich aber doch offenbart. Die Textstelle enthält aber auch *Program code*-Elemente, wobei der Punkt zwischen Body und Skin bedeutet, dass das Objekt Body die Eigenschaft Skin besitzt. Es fehlt allerdings noch eine Anweisung, zum

Beispiel die Zuweisung einer Farbe: body.colour=blue. Memmott verwendet Javascript und DHTML, es handelt sich dennoch um Pseudocode, der zwar an diese Sprachen erinnert, aber nicht ganz korrekt ist.

Wie bei Mez manifestiert sich der *Program code* nicht zuletzt in der Verfremdung von Wörtern, zum Beispiel steht "cell.f" oder "cell…(f)" für "self". Das Ich wird häufig als "I-terminal" (auch: eye-terminal) bezeichnet und dem "x-terminal" (=computer) gegenübergestellt. Das I/eye ist zentral, immer wieder erscheinen am Bildschirm digitalisierte Augen, die uns gewissermaßen aus dem Inneren des Bildschirms heraus betrachten, aber auch Reflexion unserer eigenen Augen sein könnten, also gegenseitige Betrachtung anzeigen.

Anspielungen auf Kunst, Literatur und Mythen durchziehen den Text. Freud, Nietzsche, Heidegger, Deleuze und Guattari sind häufige Referenzautoren. Eine Anspielung auf Freuds *Unbehagen in der Kultur* (Civilisation and its Discontents) steckt zum Beispiel in der Zeile "Cyborganization and its Dys ı Content(s)". Freuds Behauptung, der Mensch strebe danach, mit der Mutter zu einer Einheit zu verschmelzen, wird auf die digitale Technologie übertragen, die mit dem Menschen verschmilzt und nun als ein notwendiger, aber verhängnisvoller kultureller Entwicklungsschritt erscheint. Der Mythos von Narziss und Echo wiederum wird im Zusammenhang mit den Augen herbeizitiert, die als Augen des Betrachters, der sich somit selbst bespiegelt, interpretiert werden können. Für die postmoderne Literaturtheorie von Bedeutung ist der Umstand, dass Echo nur wiederholen kann, was andere gesagt haben. Auch *Lexia to Perplexia* ist durchzogen von solchen Wiederholungen und Echos. Schließlich wird der ägyptische Isis und Osiris-Mythos bemüht. Memmott erläutert in einem Interview, dass Osiris auch Ausere genannt wird, was nun wiederum als 'a user' gelesen werden kann. Über diesen Mythos kehrt die Thematik des Körpers, seines Verhältnisses zur Seele, von Tod, Einbalsamierung und Sublimation (Vergeistigung) wieder, die die Auferstehung des users als virtuelles Techno-Subjekt impliziert.