

# diálogo





# Egroup Webartery: diálogo e criação

---

J. ANDREWS • R. DRAKE • S. KATZ  
T. MEMMOTT • MEZ (MARIE-ANNE BREEZE)  
A. SONDHEIM • T. WARNELL

J. LUIZ ANTONIO (org.)

**Resumo** Trata-se de uma antologia de emails do Egroup Webartery, do Canadá, no período de 12 a 26 de novembro de 2000, enfeitada sob o tema *code poetry*, cujo resultado, para os leitores e para os participantes, é a riqueza de informações e a criação de *email poems* por Ted Warnell, Alan Sondheim e Mez.

**Palavras-chave** códigos poéticos, ciberpoesia, máquinas de linguagem, Internet, Egroup, Email Poetry

**Abstract** The present dialogue is an anthology of emails from November 12<sup>th</sup> to 26<sup>th</sup>, 2000, under the theme of *code poetry*. These emails make part of the egroup Webartery, from Canada, which owner is Jim Andrews and moderator is Mez. The result of the discussion, for the readers and for the egroup participants, is the richness of information and also the creation of *email poems* by Ted Warnell, Alan Sondheim and Mez.

**Key words** code poetry, cyberpoetry, language machines, Internet, Egroup, Email Poetry

## INTRODUÇÃO

Esta introdução não pretende exaltar as qualidades da Internet em detrimento de outras formas de comunicação, como o correio, o telefone, o fac símile, a conversa *tête-à-tête*, mas, sim, mostrar o resultado de um grupo de discussão — o Webartery — em que seus membros aproveitam todas as oportunidades para criar. De certa forma, é também nosso objetivo compartilhar com os leitores deste periódico as oportunidades de pesquisa que um *egroup* pode nos trazer.

O tema selecionado é o do *code poetry*, com os inúmeros significados que os membros do Webartery deram a ele, desde a mensagem cifrada da Segunda Guerra Mundial ao uso de textos de *e-mail* em HTML ou linguagem de programação. Oito pessoas participam de um diálogo num tempo cronológico que vai de 12.11.2000 a 26.11.2000. A coletânea inicial permitiu três agrupamentos semelhantes a este que apresentamos.

As mensagens de *egroups* são cápsulas de ensaios onde as conversas se misturam com pequenos textos teóricos. Há uma intencional ausência de imagens, com exceção dos frames-propagandas dos patrocinadores, e grande parte das referências são urls dos sites tanto dos participantes como de outros artistas.

Procuramos preservar o diálogo da mesma forma que ele ocorreu, na mesma seqüência que apareceu no computador.

O Webartery (<http://webartery.com>) é um *egroup* fundado no Canadá em 15 de dezembro de 1998, em língua inglesa, sob a coordenação de Jim Andrews, atualmente com cerca de 130 membros de muitos países, moderado pela australiana Mez, cujo objetivo é discutir Web Art ou Net Art, o que envolve a síntese de artes previamente separadas: escrita, som gravado, trabalho visual, etc. O mais abrangente propósito da lista é participar de um pensamento através das poéticas da web art. Todos os emails desse *egroup* são catalogados desde 1999 até o presente momento no site de Jim Andrews como uma grande obra coletiva, sob o título de *Webartery Email Correspondence* (1999-present). Para se ter uma idéia, até o dia 25.06.2001, havia 11.950 emails trocados entre seus membros.

No final do trabalho encontra-se o *email poem*, elaborado por Ted Warnell, com várias colaborações. É interessante observar que a forma e o conteúdo se assemelham a tudo quanto podemos considerar como poesia verbal, sonora e visual no contexto eletrônico-digital. A imagem se faz através da disposição gráfico-espacial das palavras, as palavras são comandos de programação ou linguagem HTML.

Agradecemos a todos aqueles que gentilmente autorizaram a publicar os seus diálogos.

---

Ted Warnell — 11/12/2000 — `::codePoetry]`// — The term 'code poetry' goes back a ways. From what I am able to find on the Web, at AltaVista and the snappy new, totally useful and beautiful (and FAST) Google.com search engines, the term has long time but limited definition as:

- 1) literal, as in poetry realized in Morse Code.
- 2) literal, as used to convey covert messages in time of war — (there are several examples of this — interesting use of poetry).
- 3) some refer to the 'art of (digital) programming', and the resulting computer code as 'poetry'.
- 4) there are a couple of references, titles of vizpo works, I think, as CODE POEMS, dating to about 1968.

Found also a single reference by Margie L., at ELO chat archives, which reference, judging by the context of the conversation, is as we here at webA are most familiar: a genre of poem/poetics. Also, there are numerous references here in the webA archives that go to this idea of a genre of poem/poetics, and in the Glazier Defib.

Beyond these, there is little else on the subject of code poetry.

When I think of code poetry, I immediately think of some of the lively, often spontaneous exchanges that happen frequently here at webA — Brian Lennon, Mez, Jim Andrews, Tom Bell, et al. Also, of specific works by these artists and others — Steve Duffy, Thuan Tran, Joe Keenan, Alan Sondheim, Reiner Strasser, and etc.

I am aware, too, that creative (poetic) email exchanges between artists has been going on for some time, though my knowledge of these exchanges is limited. My sense is that they are not too interested by code poetry (as we know/practice it here) as much as by other concerns (no less creative and interesting than ours).

I am aware of various 'other language' (ASCII, mathematics) and machine generated/modulated poems by artists like Jim Rosenberg, John Cayley, Jackson Mac Low, and numerous others. These works, too, strike me as less interested by code poetry than by other things.

I am wondering if there are thoughts about code poetry as practiced and known here at webA — if it might be possible to describe some of the more prominent features of the code poem without need or fear of being 'prescriptive' about it. I'd just like to hear thoughts of other practitioners about how you see it, what it means to you.

To get this ball rolling, I will say:

Code poetry is concerned by interactivity (real-time, live, exchange, performance vs Web typical point-and-click pop-up/jump);

Code poetry is concerned by structure, often, utilizing capabilities of fixed-width (non-proportional) fonts to achieve specific layout and presentation;

Code poetry is a dynamic process, often, with multiple 'authors' engaged in forming and re-forming 'a work' — many minds, evolving ideas taken (simultaneously) in different directions;

Code poetry is concerned with 'language' and the stuff thereof, like syntax and punctuation — a new visual language, symbolic rather than literal — often multiple meanings — non-narrative — poetic.

The term 'code poetry' is not literal, not merely poetic, not a title — it is something else/more/less, such as...

Ted Warnell — 11/12/2000 — ::codePoetry]//2<sup>1</sup> - Code poetry//the process:

Usually is dynamic — decidedly 'new media' (Net dependent).

Code poetry is concerned in a big way with things cyber (digital), as both processes (email, IRC, etc.) and the sequences, commands, code requirements/structures of same, i.e., machine languages.

Code poetry//the artifact::

Strong visual component (re structure/layout/presentation) seems to owe something, then, to concrete poetry, perhaps. Presentation of the artifact may or may not require new media; as a simple text it might as easily be presented in print as on the Web, but it might also be presented as interactive/dynamic/otherwise Net-embellished 'new' work beyond literal text.

Code poetry is often 'self-referencial' in that it often will attempt to 'make visible' underlying supportive code/coding structure of it's makeup/existence/fact. The underlying 'neath code' (Jim's term, nice) is sometimes central to the concept/structure/realization of the code poem proper, but not absolutely always.

other thoughts...

Revere 1955 (Susan Katz) — 11/13/2000 — ::codePoetry]//2 — other thoughts..

I am still listening intently, Ted, as you search for the defining thoughts.

Susan :)

Jim Andrews — 11/13/2000 — ::codePoetry]//2 — Great posts, Ted. I think you

really opened it up, explored it fairly comprehensively in your posts, at least in suggesting a wide range of properties and concerns of 'code poetry'.

I don't know what poetry is, but it seems that it's intensely engaged with language.

1. Durante um dia, por exemplo, muitos participantes escrevem vários emails.

Computers are fundamentally language machines. Some would say no, they're number machines. Yes, in a sense, but the numbers (or configurations of transistors) are placed under an interpretation. At its most basic level, the state of a transistor (on or off) is interpreted to 'be' or 'represent' a 1 or 0. There are no 1's and no 0's in the computer; a look inside the box assures us of this.

At a slightly higher level, a sequence of 8 transistors, capable of being in 2 to the eighth power (256) different configurations (on on off off on on off on)=(11001101), (on on on on off off off off)=(11110000) is the basic unit (a byte of information), by consensus, of humanly relevant information, since 256 possibilities covers the range of symbols in the western alphabet and numeric system; computers were developed in the west. At this slightly higher level, (11001101)='g' or some other letter or number or arithmetic operator such as '+' or whatever the ASCII correspondent of (11001101) is. One might say that there are other codes than ASCII and this is true but would only emphasize my point that the computer is not so essentially a number machine as a language machine; whether the code is ASCII or something else, an interpretation of the 'meaning' of the sequences of transistors is necessary, ie, an imposition upon the machine's mindless, concrete states of language is fundamentally necessary.

Computers are fundamentally language machines in this sense and in other senses. It is not an overstatement to say that the great mathematical achievements since Shannon's information theory in the forties have typically been concerned with language. It's interesting to note that some of Noam Chomsky's work in linguistics is fundamental to computer science and is studied by most third year students of computer science in a course that covers 'language and the theory of computation'. Chomsky was originally trained as a mathematician. This course also typically covers some of the work of Turing and Godel.

Of course, the breaking of the Nazi encryption coding of messages was a crucial impetus to the work of Turing and others. The development of computers has been very intensely concerned with language, as is poetry.

Mathematics, previously, was not much concerned with language. The exemplars of applied mathematics previous to the computer age were in physics, where matter and its fundamental components and properties and dynamics were the primary focus, rather than the current situation where it is information that is the primary focus.

So I see 'code poetry' as being in relation to the contemporary shift in scientific/philosophical concentration on information and language. Finally science is addressing language in a profound way. To explore 'code poetry' is, in part, to

contemplate the views of language and ourselves that are arising out of the synthesis of approaches to language, computation, matter, and mind going on in science and communications.

J.

---

Revere1955 (Susan Katz) — 11/13/2000 — `::codePoetry]`//2 — To explore 'code poetry' is, in part, to contemplate the views of language and ourselves that are arising out of the synthesis of approaches to language, computation, matter, and mind going on in science and communications.

very good, Jim.. now we're reaching synapse..

*"Touched by poetry, language is more fully language and at the same time is no longer language: it is a poem."* ~Octavio Paz

Susan

---

Jim Andrews — 11/13/2000 — `::codePoetry]`//2 — Nice quote. Another:

*"A poem is a machine made out of words."* ~William Carlos Williams.

I used this quotation in a piece at <http://vispo.com/animisms/enigman/EnigmanInfoanimism.html> which, though not explicitly about 'code poetry' is about my own particular approach to similar, related territory, as are many of the essays in <http://vispo.com/writings>.

J.

---

Jim Andrews — 11/13/2000 — `::codePoetry]`//2 — Also, Bill Marsh has written a nice essay on hypermedia and the timeline ("Reading Time: Duration and Temporality in Digital Writing" at <http://sunbrella.net/btheater/works/essays/ht/time/text.html>). I believe he is contemplating a related essay on the flowchart. The notion of the timeline and the flowchart seem related to 'code poetry'. Looking forward to it.

For a rather defensive recent piece on 'cyber poetry', have a look at Brian Stefan's piece at [http://www.ubu.com/feature/papers/feature\\_ol.html](http://www.ubu.com/feature/papers/feature_ol.html). Gotta say I chuckled at that one. I've invited Brian to webartery but I see his fondness for the Poetics list in that essay. He is often a brilliant guy but it seems his goat has been gotten in that essay somewhere along the line.

And Alan and Talan have been doing great work writing about not maybe 'code poetry' specifically but related. Loss too; he has an essay in DOC(K)S called "ABC's of Coding". And many others.

Domain name: 'code poetry'



'Code poetry' is also a great pogoat getter. But in the main, it does seem that little has been written on code poetry, yes, compared with, say, hypertext.

J.

---

R. Drake (Luigi-Bob Drake) — 11/13/2000 — ::codePoetry//2 — close, Jim"

*"To make two bald statements: There's nothing sentimental about a machine, and: **A poem is a small (or large) machine made of words...** Prose may carry a load of ill-defined matter like a ship. But poetry is the machine which drives it, pruned to a perfect economy. As in all machines its movement is intrinsic, undulant, a physical more than a literary character... When a man makes a poem, makes it, mind you, he takes words as he finds them interrelated about him and composes them—without distortion which would mar their exact significances—into an intense expression of his perceptions and ardors that they may constitute a revelation in the speech that he uses. It isn't what he says that counts as a work of art, it's what he makes."*—William Carlos Williams, from "Authors Introduction to *The Wedge*"

---

Ted Warnell — 11/13/2000 — ::codePoetry//2 — Interesting quote (both Jim's and LBD's)

I like Williams' thoughts in this fuller quotation — with thanks, Luigi. Ted

---

Peter Howard — 11/13/2000 — ::codePoetry//2 — On Sun, 12 Nov 2000, Jim Andrews wrote...

Computers are fundamentally language machines.

I can't agree with this, Jim. Your argument seems to be based on the fact that a byte can represent a western character. Yes, but it can also represent a number, a shape, or anything else. You might as well say that computers are fundamentally musical machines, because the information could represent a note.

Certainly, information in binary form needs interpretation. But interpretation is not the same thing as language. One needs to define what one means by a language before taking about it. Otherwise one ends up in a definitional mire. Of course people mean different things by the word, but one important characteristic of most definitions is symmetry. Computer 'languages' or linguistic applications of computers generally lack this. What I mean is that, although I might make my computer talk to me in English, in order to do so, I don't talk to it in English, but in C++ or something.

It is not an overstatement to say that the great mathematical >achievements

since Shannon's information theory in the forties have typically been concerned with language.

Yes it is. Two of the most important have been catastrophe theory and chaos theory, neither of which are concerned with language.

An interesting post, Jim, but I think you're claiming a little too much here.

Best, Peter

---

flesh h][i.j]acker (Mez) — 11/13/2000 — ::codePoetry]//2 — At 08:59 AM 13/11/00 -0600, RDrake wrote:

"When a man makes a poem, makes it, mind you, he takes words as he finds them interrelated about him and composes them..."

...and a woman?

---

Ted Warnell — 11/13/2000 — ::codePoetry]//2 — No, Mez, a man does not compose a woman.

Revere 1955 (Susan Katz) — 11/13/2000 — ::codePoetry]//2 —

...and a woman?

just what I had been thinking...

funny, Ted.

right, Mez.

oh, Ted.

Susan :)

*"The cricket's gone, we only hear machines; In erg and atom they exact their pay. And life is largely lived on silver screens."* David McCord, Ballade of Time and Space [1935]

---

Ted Warnell — 11/13/2000 — ::codePoetry]//2

No one here, Mez...

---

flesh h][i.j]acker (Mez) — 11/13/2000 — ::codePoetry]//2 — At 02:21 PM 13/11/00, you wrote:

No one here, Mez...

hopefully, Ted... yes, and I land largely on your side this morning...

a shame though that we can't be indicated, even through a slim-lined quote such as the contentious, marked by that biological cross...[and, I know, let it slide, take the gloss, be a good sport, etc] ah well mebbe the required dose of radiohead has done strange thangs to my noggin....

in flesh boundaried,

Mez

---

flesh h][i.j]acker (Mez) — 11/13/2000 — co][deP][I][oetry] — At 02:21 PM 13/11/00, you wrote:

No one here, Mez...also, I didn't mean to point the indication finger, as in make it case [webA] specific, more throwing it back out there in2 the tumultuous ether...

[read: a distance broken, eyes alert to changes in the sense-oh-sphere, dips and waning cusps...unveiling and skeleton paring to the core?]

[a direction melt is mono in its .man.ifest gaze, keeping a focus on a relevant wire doll frame...posting in that orange fused g][I][aze].....]

[ah, the crushing w][eight][][t][ai][n][t, a pink-dressed child's mouth intent on trembles, my vipers arm dissected into threads, uterus eye-red and .man.a cord cut, my muse is fe][mme][tal, replete, concussed if othered, gender gelling in2 clumps as thin red drainpoints blossoms in2 dawn....]

[heed point terminal. critica][u][I .man.ouvre.ing. bent knee&head at the jolt of an.other. ][deep][. total organ failure likely, and, as always, hysterical.]

[invert intent and deploy.]

well.

at least it promotes growth.

---

flesh h][i.j]acker (Mez) — 11/13/2000 — co][deP][I][oetry] — didn't mean the triptych, not sure how that happened, am tempted to apologize but mebbe it was following another ][more salient][ trajectory altogether...

[read: sorry on a need-to-grab basis.]

---

Alan Sondheim — 11/13/2000 — codePoetry — As usual apologies; I'm on the road, my connection is intermittent, and I have tried for the past half hour to read until the end of the digest, but haven't been able to — I have had to crash out of stalled accounts, etc.

Anyway, I would think that "codePoetry" is really referencing, at least in my case and maybe NN's for example, the underlying strata of protocols sed online — sometimes these are part and parcel of surface content, and sometimes they're substructural. Think of surface content as parasitic or lesh (Niku) covering the bones or workings of things. This email for example rides upon numerous protocol layers.

In addition, one might say that codePoetry rides among or between layers — sublayers, layers, and metalayers — and even among or between subnets, nets, and internets.

I don't see it as a style so much as an investigation, perhaps related to the subsumption architectures of Brooks, which are used as well by Brian Rotman in his work on mathematics – architectures in which the environment is primary, so that what is visible as symptom is only a very small part of the connectivity. Maybe subsumption literature – or to use Aarseth's term in another sense – subsumption ergodic literature – would make more sense...

Alan, apologies for the scribbling, can't see what I'm writing here -

---

Talan Memmott – 11/13/2000 – codePoetry – Hittin' it on the head here, Alan....

Some of what you mention here plays into my work, as you know...

More and more so, as I work through more models for the apparatus as the bracketed set [agency|appliance|apparatus].

I think Lexia to Perplexia makes some statements/examples/experiences through language and functionality that operate at the level of severe subsumption... The 'ergodica' is harder to put your head around and metabolize....

---

Ted Warnell – 11/13/2000 – ::codePoetry]//3 – This poem was written by British Special Operations codemaster Leo Marks, and was used by WWII agent Violette Szabo as a code poem during a mission in occupied France during the second world war:

Yours

The life that I have is all that I have,  
And the life that I have is yours.

The love that I have of the life that I have,  
Is yours and yours and yours.

The sleep I shall have, a rest I shall have,  
Yet death will be but a pause.

For the peace of my years in the long green grass,  
Will be yours and yours and yours.

Yours is one of hundreds of poems written by Marks for use as a code cypher.

Violette Szabo was executed as a spy in Nazi Germany in 1945 with two other women SOE agents, Lillian Rolfe and Denise Bloch, and was posthumously awarded the British George Cross, and French Croix de Guerre in 1947.

]//

---

revere1955 (Susan Katz) — 11/14/2000 — ::codePoetry]//3 — this is beautifully written as is, but I'm curious to know the decoding.

Susan

Ted Warnell — 11/14/2000 — ::codePoetry]//3 — Hmm, yes — it is a lovely poem — really quite beautiful especially considering that it is really just a tool, a code cypher. Didn't find any more info about how it works, and really don't know anything about the subject, Susan, but I'd guess if you wanted to look at cryptography and the encoding/decoding of messages, there you'll find it.

Ted

---

Ted Warnell — 11/14/2000 — codePoetry]//2 — if one were to want to try the form you mention below, Peter, Lingo would not be a bad language to try it in. Recently they have introduced the familiar dot syntax (sound(1).volume for instance is the volume of sound channel 1) but previously it was quite a 'verbose' language so that you could say 'set the volume of soundchannel1 to 255' and have it compile. 'the' and 'of' and other such articles and connectors are a part of lingo.

I have seen some pretty good such poems from Ted in javascript, actually, though I don't know whether they were meant to compile (or 'be interpreted' since most of the languages we use are interpreted rather than compiled).

On the matter of computers as language machines (or not), Peter, I read of MIT students making a 'computer' out of mechano sets. Probably not exactly a pentium, but a computer nonetheless, the point being that computers are not so much made out of silicon and transistors etc as from logic. A Turing machine, and all computers are Turing machines, is a mathematical model of the essentials of a computer. It isn't actually a physical machine. It is an abstraction. And the fundamental thing of a Turing machine is that it 'reads' symbols and 'writes' symbols, it manipulates symbols, it deals with 'language' fundamentally, whether the language is 'musical' or whatever.

In the Turing machine, the fundamental model of all computers, the notion of 'reading' and 'writing' and manipulation of symbols that form various types of languages, is present. So I still think it's accurate to describe them as language machines.

On chaos theory and catastrophe theory, I know a bit about chaos theory, not much (from “complex analysis”). Nothing about catastrophe theory. And there’s also Fermat’s last theorem solved now, which I doubt has much to do with language either. So, yeah, there’s undoubtedly a lot of high profile contemporary work in mathematics that doesn’t deal with language.

It has been said that the human ability to manipulate symbols to the degree that we can is one of the distinguishing characteristics of humans, among other beasties. Seems plausible though who knows what those crows are talking about (besides ‘watch it buster, I’m a very bad bird’). The immense flexibility of the computer, flexibility to the point where it’s an open question as to whether it is capable of thinking, is owing to its ability to manipulate symbols also.

Code poetry, to me, is involved in this mystery of the relation between language and computation.

Jim

---

Ted Warnell — 11/14/2000 — ::codePoetry//2 — Jim writes to Peter: Code poetry, to me, is involved in this mystery of the relation between language and computation. Would agree with this, Jim, about mystery — from winsock startup on my PC these traces indicating there is indeed magic in the computations...

```

PPP[C021] state = starting
PPP[C023] state = starting
PPP[8021] state = starting
PPP ENABLED
PPP[C021] SND CONFREQ ID=01 LEN=24 MRU(0400) ACCM(00000000)
MAGIC(00511893) PFC ACFC
PPP[C021] state = reqsent
PPP[C021] RCV CONFREQ ID=01 LEN=24 MRU(0400) ACCM(00000000)
MAGIC(00511893) PFC ACFC
PPP[C021] SND CONFACK ID=01 LEN=24 MRU(0400) ACCM(00000000)
MAGIC(00511893) PFC ACFC
PPP[C021] state = acksent
PPP[C021] RCV CONFACK ID=01 LEN=24 MRU(0400) ACCM(00000000)
MAGIC(00511893) PFC ACFC
PPP[C021] state = opened
PPP[C023] SND CONFREQ ID=01 LEN=21 data 07 77 61 72 6E 65 6C 6C
08 7A 65 6E 6F 67 69 6D 70
PPP[C023] state = reqsent
PPP[C023] RCV CONFREQ ID=01 LEN=21 data 07 77 61 72 6E 65 6C 6C
08 7A 65 6E 6F 67 69 6D 70
  
```

```

PPP[C023] SND CONFACK ID=01 LEN=4
PPP[C023] state = acksent
PPP[C023] RCV CONFACK ID=01 LEN=4
Accepted:
PPP[C023] state = opened
PPP[8021] SND CONFREQ ID=01 LEN=10 IPADDR(C7B9E1D4)
PPP[8021] state = reqsent
PPP[8021] RCV CONFREQ ID=01 LEN=10 IPADDR(C7B9E1D4)
PPP[8021] SND CONFACK ID=01 LEN=10 IPADDR(C7B9E1D4)
PPP[8021] state = acksent
PPP[8021] RCV CONFACK ID=01 LEN=10 IPADDR(C7B9E1D4)
PPP[8021] state = opened

```

Point to Point Protocol.  
Point is Poetry Protocol?

l a n g u a g  
e l a n g u a g e

elanguage

"Touched by poetry,  
language is more fully language  
and at the same time is no longer language:  
it is a poem."

— Octavio Paz

[ thx S ]

Is language any less language for being non-verbal?  
Is language any less language for being un-written?

---

Jorge Luiz Antonio — 11/14/2000 — Open Do Down — Jim, I have been collecting all emails about code poetry. It seems very useful for me, for my doctorating thesis, as a kind of thinking about poetry in computer, that is, by, through, with computer.

To find a language which situates between verbal, visual and sound language or syntax and machine one, that is what we have been looking and searching for.

My thesis is that we have already found the way although we have been searching new other ways to do so, at the same time technology increases different ways of using a computer.

That's what I would like to say about code poetry for the moment.

Thank you. Jorge Luiz Antonio

---

Jim Andrews — 11/14/2000 — Oppen Do Down — Thanks, Jorge. Yes, you're right.

The sounds in this piece are the result of my initial stumblings around trying to compose sounds that are interesting when layered one on top of another, not so much one after another.

Reiner Strasser and some others here, but particularly Reiner, has written about layers as one of the interesting dimensions of what we deal with in web.art/net.work etc. Visually but now also sonically. And conceptually. And in the code below the page.

I ended up with fourteen sounds but only a few of them sound good together. So I thought what am I going to do with this mess? instead of just presenting the three or four sounds that sound good together, I thought, well, if I present them all and allow them all to be heard or not, however the user wants, that is in a sense more important than just presenting the ones that sound good since this way it shows more of the exploration I made in making the sounds and trying to think about layers of sound. It is more interesting as an exploration rather than a polished finished thing, rather than just a sort of song, I hope it shows more of the directions I tried to explore, some successfully, some not.

I hope I learned from it. The coding in this piece is actually quite simple compared with the other vismu pieces I've done. This is because this piece really only deals in layers of sounds rather than dealing with sequences of sounds also.

Jim

---

Jim Andrews — 11/14/2000 — Oppen Do Down — In progress: Oppen Do Down.

needs shockwave and a sound card capable of 8 channels of sound.

Wish I had a better falsetto, yikes.

if you play more than one sound and they get out of synch, click 'synch':

mostly the sounds were meant to mix with 'badly' though not all.

an odd sound poem; <http://vispo.com/vismu/OppenDoDown.htm>

---

Jim Andrews — 11/15/2001 — ::codepoetry][//2 — I think our disagreement

over whether computers are language machines is largely definitional: I have a stricter view of what constitutes a language than the one you're using. But I do think your less rigid definition runs the risk of attributing more to what computers are about than is actually the case.

Non Euclidean geometry loosened up the idea of what a straight line is: simply the shortest distance between two points. In a Euclidean space, a 'straight line' is our intuitive notion of it, then, but in a warped space, the shortest distance between two points is not always 'straight' in the 'straight' sense of 'straight'.



Similarly, the notion of what a language is might require some generalization, Peter, to place poetics within contemporary thought about language.

Looking at "Introduction to Languages and the Theory of Computation" by John Martin, for instance, I see that a 'language' is described simply as a particular set of strings over a particular alphabet.

This is the sort of context in which computation and language relate.

Regards, Jim

Jorge Luiz Antonio – 11/15/2000- ::codePoetry]//2 – Jim, It is very interesting the discussion about code poetry and the comments regarding Brian Kim Stefan's article "Reflections on Cyberpoetry".

I have been following all the commentaries and it has been enlarging my viewpoint about web/electronic/digital art/poetry.

As I could not read all the Brian's articles up to know, it seems that "cyberpoetry" is something like "digital poetry", "computer poetry", "infopoetry". Is that right? Is not there any different meaning for "cyberpoetry" which I could not get? Well, up to the point I have read, Brian intends to say the "cyberpoetry" does not exist because it not revolutionary yet. From this point of view, he has forgotten that we do not have yet historical perspective.

But, anyway, the discussion is becoming very interesting more and more.

Best regards, Jorge Luiz Antonio

Jim Andrews – 11/24/2000 – code poetry – <http://vispo.com/animisms> has what i would consider some code poetry in there: Enigma n, Seattle Drift, Millenium Lyric, Time Piece, the Stir Fry texts. The Vismu work certainly has a lot of code in it.

But what Ted means by 'code poetry' is probably a bit different than what I'd define it as. But hopefully the idea could accomodate us both.

Code can be on the screen or beneath the page or both or implicit in the text but not 'seen' as in a secret message.

The Infoanimism essay I wrote, which is related to Digital Langu(im)age at <http://vispo.com/animisms/enigman/EnigmanInfoanimism.html> addresses a notion of code poetry, in a sense.

And <http://vispo.com/writings> has several essays in it. Stir Frys and Cut Ups does a bit of that.

Loss Glazier has written a bit about code poetry. His site is at <http://wings.buffalo.edu/epc/authors/glazier/> and stuff on code is at <http://wings.buffalo.edu/epc/authors/glazier/essays/>

Bill Marsh has some interesting writings sort of on the subject at [\\_http://sunbrella.net/btheater/essays.html](http://sunbrella.net/btheater/essays.html)  
I'm working on an essay right now that is kind of related:

## INTERACTIVITY IN DIGITAL ART AND WITH COMPUTERS MORE GENERALLY ALWAYS ALREADY INTERACTIVE

Just about everything is always already interactive; the issue of interactivity is very broad. Books are interactive: meaning is not something ready made in writing but is constructed. We ourselves are interactive as opposed, say, to being passive receptacles of information that programs our actions accordingly. Computers are such things, though of course it's an open question as to whether you can program a computer in such a way that it can think.

What interests me in new media works concerning interactivity are works in which there is a mixture of media and ways to interact with the work that create a kind of dialogue between myself and the author, or creates a dialogue but confuses just exactly who the dialogue is with, only the 'language' of this dialogue is via the choices presented and the consequences of those choices. Not to say that the author should know the consequences of all the choices permitted, just as the programmer of an application like a word processor does not know what the user is going to write.

## INTERACTIVITY IN APPLICATIONS AND ART

However, there would seem to be a difference between the interactivity I'm talking about in works of art versus the interactivity set up by programmers of applications like word processors. In an application, the content is typically almost entirely up to the user. Like in a word processor. Or in a sound sequencer. Whereas in applications that blur the line between the application and a work of art, there is typically more content supplied by the author/programmer and the application allows people to navigate and also manipulate the content in various ways. Some of which the programmer/author can't predict. But writers are used to a situation where they cannot predict what people will make of their work anyway. And often, the user's control over the manipulation of the content is not complete; the content resists complete control by the user, unlike the non-buggy application (and where is that?).

Also, in a strict application, the programmer wants to follow the conventions of menu design and other conventions of interface design quite thoroughly in order that the user be able to bring their past experience of how applications work to bear on the current application. The idea being that if you know how to use several applications, you basically know how to use them all. So that the interactivity is as standardized as possible into the transparent language of interface design. After all, the goal is to let the user get on quickly with making whatever sorts of content the application is designed to allow them to make, not talk with the user or differ with them concerning interface design. The goal is to get out of the way completely and let them make what they will.

Whereas in a work of art, though the author may leave much up to the imagination, and probably should, there is nonetheless usually some (uh, how do I say this?) content to be processed by the user rather than the work of art simply processing the input from the user.

#### UNPROVABILITY AND APPLICATIONS AS BARTHEAN ART

A programmer might reply that there is content to be processed by the user anyway in an application: how to use the application and understand the possibilities it offers. And I think that would be hard to argue with. And is the basis of the possibility of blurring the line between the application and the work of art, or part of the basis. Another principal being that it's good in art to leave as much as possible up to the imagination of the reader. Certainly applications leave a great deal up to the imagination of the user. The application as Duchampian ready made. Neither beautiful nor ugly. Eliciting no particular aesthetic response from the user. Yet very carefully made to permit the user's imagination full reign toward the making of something themselves, making something out of the application itself.

This is rather atypical 'art' though, if it is art at all. On that question I tend to agree with Joseph Kosuth who says that if somebody calls something art then it must be art, for to assume to the contrary that there is some sensible way to go about defining art so that you can determine if something is art or not is finally preposterous and certainly not worthy of art itself. And it is prescriptive in a way that shuts down possibilities which, to me, anyway, is contrary to the nature of art and good art criticism. Saying 'X is art' is a bit like saying 'This proposition is not provable' since one arrives at a contradiction should one say it's false; no contradiction seems to arise if one says it is true, yet neither then is it provably true. Nor

is it an axiom completely independent of the rest of the language system, for if it were, its negation would be an acceptable alternative to the proposition itself, but it is not. It is an unprovable statement that nonetheless must be accepted as true. Each new piece of 'art' is thus kind of like an axiom rather than a theorem.

A programmer might go on to say that applications can be quite innovative in their use of language, namely the language of interface design, which is highly visual in its nature and by now is probably one of the most international languages going.

One might reply to the contrary that rather than being innovative with nterface design, it is in the interests of programmers and ease of use to make the design as standard as possible and reserve the innovation for the features made available to the user, and possibly the interface components of those, if there are simply no iconic/organizational precedents concerning these features and their use.

One does not expect or really want the programmer or the program intruding in on the use of the tool. One does not wish the wrench to mutter between clenched teeth or, worse, declaim on the wrenching art. We will do it ourselves as we please, thank you very much.

The programmer might reply that at least she can use the language of the application interface with wit or howsoever she pleaseth, whereas digital art that does not aspire to even a little bit of the application misses out on using this language utterly, which after all is the primary definitional language concerning 'computer literacy'-if poetry is intensely involved in language and should have at least an inkling about the medium it occupies, then surely this language is of relevance-look at what's in front of your nose, god damn it, look at what you're doing, she says, and adds 'if we should speak of computers and interactivity, this 'language' is fundamental to it' Right on, grrl.

The application is typically a tool, however subtly it shapes our ends, as tools do. The interactivity is geared toward letting us make what we make with the tool easily, intuitively, and the tool should have all the totally selfless and immediately serviceable characteristics of useful tools. The art is in designing dynamite features and in preparing the environment suggestively and then getting out of the way completely, disappearing from the interaction. It may be art, if you like, but if so it's Duchampian or Barthean. I like it. Digital art that uses the application or essentially is an application is often not so nice, either; it threatens to take over your machine and really get in your face. The brilliant work of Antiorp comes to mind in this regard.

INTERACTION WITH A TOOL OR WITH PEOPLE?

Interaction in multimedia works of art on the computer may incorporate the application in these ways, may be tools that the wreader uses to create texts or images or sounds or whatever or may use the piece to communicate via IRC or whatever, but typically that is not all they do, though, as above, that could surely be sufficient. Typically the author does not disappear utterly from the interaction, or confuses the matter of authorship, and one feels there is some sort of non-mechanical dialog going on with the author or the piece itself or the subject matter of the piece.

The interactivity speaks a subtle and engaging narrative of revelatory possibilities. The 'content' of the work is typically not as entirely up to the 'user' but there is enough of it that the term 'reader' comes to mind and then also the 'wreader' (the reader/writer) rather than the 'user'.

If writing and painting etc are always already interactive, then the reader or observer is always already creatively interpreting the 'meaning' of the work, constructing her/his own interpretation of what the work is about and is saying, for instance, or constructing a frame for the work, ie, putting it in larger contexts.

When we use a word processor, we do not typically engage with it in this way. We are writing a friend or writing this essay right now on a word processor which is not so much about word processors as about the processing of words and interactivity, about the various ways we read and write.

On further reflection, we see that when we engage with a work of art we are not just engaging with it but, rather, in our creative process of interpretation and framing of the work, we are sometimes talking to a friend literally or in our heads, we are using the work of art as a stimulus or gathering point or launching board for thoughts and feelings concerning our own lives or ideas, loves and so forth in a manner similar to the way in which we use a word processor. Which is as a tool, primarily.

Good interactivity in multimedia art can provoke questioning about the dynamics of interactivity. Who or what are we interacting with? Is there an author here? Am I the author? Is this a tool or not? What is a tool? I'm operating on this piece; how is it operating on me?

Also, often there is a certain amount of interactive creation of the content, or manipulation of the content into widely permutative new configurations. Typically, in digital art, the range of this combinatorial complexity is not as wide as it is in a word processor, say. And it isn't meant to be. The content exists between being in a fixed state and being totally created by the wreader. The wreader creates some sort

of permutation or derangement of the existing content. So that the art work is not a fixed and singular thing, but neither is it as totally up to the user's creation as typically happens with using a word processor. It is as though there is a 'range of readings' available, or a range of works themselves within the one work. But really that range defines one work, usually. The 'meaning' of the work should be considered amid the total possible configurations. The ranges of meaning possible within the various configurations, in sum, speak to the meaning of the whole piece.

## TOOLS GALORE AND COMPUTER LITERACY

I wrote an applet a few years ago called MORPH TEA that is pretty strictly a tool. It allows people to display on their web pages animations of a certain sort fairly well. It is freely downloadable, and there have been thousands of downloads of it. And it is the primary portal for people coming to my personal site, apart from the search engines. It seems there are at least as many people looking for tools on the Web as there are people looking for art on the Web. Indeed, I suspect many people use the Web as a reference book or as a source of tools or information or consumables as opposed to using it much in dialogues of art and philosophy, for instance.

The whole use of the Web is prominently as a tool. So, personally, I find it fruitful in my art to create interaction that blurs the line between the tool and the work of art. The interaction we engage in with digital tools is fairly, um, standard. Here's a program. Here's what it does; here's what it finds or makes or otherwise allows you to do. It works like you'd expect, hopefully, given the fairly standard ways that applications work. People who are 'computer literate' are 'computer literate' to the extent that they understand the standard ways that applications work. There are of course higher levels of 'computer literacy' such as the extent to which one understands how computers or networks work. But that is esoterica compared with basic computer literacy. Using a computer involves this basic literacy developed over time. It is this type of literacy that is specific to the computer.

It \*is\* a type of literacy.

The issue of 'interactivity' in computing is an issue of computer literacy.

## BRING IT HOME

As the web.artist Ted Warnell says, "realization is the meaning". This is particularly true of computer applications where the 'meaning' of the work is very much tied to what you yourself realize with it, whether this be some sort of set of files or

whatever else you make or realize with it. But it is also true of art, for art is a kind of tool also.

So if the basic literacy with computers involves being able to use applications as tools, then basic interactivity with computers is with the machine, running the machine, getting it to do what you want it to do, getting familiar with the way that applications organize the commands available and the Help files and copy cut paste and go here no go there etc., driving the thing.

Then there's the interaction with what we make from it, with it. The documents we write, the images we make, the Web pages we read.

And also there's the interaction with other people. The email we read and write, the talking with other people via chat or net phones, etc.

But of course all of this is still, in a certain sense, interaction with the machine, however much it actually involves interaction with other people. And all too often people take the same attitude toward their dealings with people via computers as they do with their software. Get them to do something for you.

So it is important to disrupt that expectation. To be real. To take responsibility for what one says and what one does. The net is too often a sink hole of anonymity and pretence. But there are real people, always, on the other end of our missives and works. Multimedia interaction can simulate a la la land of flashing fun gadgets or it can offer deeper interaction with others, ideas, issues, philosophy, poetry, music, etc. It can open into community and real exchange, meaningful interaction, or it can be merely amusing.

In any case, I suspect that a genuine interest in exploring the ways we interact with the computer is important in being able to really implement situations where that interaction has larger significance, where the monitor is not simply a mirror of inner fantasy, but where we can experience the ways we are of each other.

---

Jim Andrews — 11/26/2000 — antiorp and code poetry — ok, i got the poop on nato055, never mind.

Here is antiorp's writing on her variant of 'code poetry': Very well done, really. <http://www.eusocial.com/nato.0+55+3d/242.ircam>.

R. Drake — 11/26/2000 — antiorp and code poetry — on 11/26/00 8:31 AM, Jim Andrews at [jim@vispo.com](mailto:jim@vispo.com) wrote:

ok, i got the poop on nato055, never mind.

Here is antiorp's writing on her variant of 'code poetry': Very well done, really. [\\_http://www.eusocial.com/nato.0+55+3d/242.ircam.html](http://www.eusocial.com/nato.0+55+3d/242.ircam.html)

be sure to view source...

Optamos por apresentar os autores através das suas urls e emails.

JIM ANDREWS – <http://www.vispo.com> – [jim@vispo.com](mailto:jim@vispo.com)  
JORGE LUIZ ANTONIO – <http://www.vispo.com/misc/BrazilianDigitalPoetry.htm> – [jlantonio@uol.com.br](mailto:jlantonio@uol.com.br)  
R. DRAKE (Luigi-Bob Drake) – [rdrake@burningpress.org](mailto:rdrake@burningpress.org) – : <http://www.burningpress.org>  
SUSAN KATZ – <http://communities.msn.com/ThePoetsPlace> – [revere1955@msn.com](mailto:revere1955@msn.com)  
TALAN MEMMOTT – <http://memmott.org/talan/> – [talan@memmott.org](mailto:talan@memmott.org)  
MEZ (Mary-Anne Breeze) – [www.hotkey.net.au/~netwurker](http://www.hotkey.net.au/~netwurker) – [netwurker@hotkey.net.au](mailto:netwurker@hotkey.net.au)  
ALAN SONDEHEIM – [http://www.anu.edu.au/english/internet\\_txt](http://www.anu.edu.au/english/internet_txt) – [sondheim@panix.com](mailto:sondheim@panix.com)  
TED WARNELL – <http://www.warnell.com/> – [warnell@memlane.com](mailto:warnell@memlane.com)

galáxia | n. 2 | 2001  
ANTONIO (ORG.), ANDREWS, DRAKE, KATZ, MEMMOTT, MEZ, SONDEHEIM, WARNELL