Approaches to Applied Semiotics
6

Mouton de Gruyter
Berlin · New York

Self-Reference in the Media

edited by
Winfried Nöth
Nina Bishara

Mouton de Gruyter
Berlin · New York

Looking through the computer screen: Self-reflexivity in net.art

Marie-Laure Ryan

As a feature of period style, self-reflexivity is generally considered a sign of old age. When it runs out of stories to tell, of new territories to explore, of discoveries to make, thought can always turn back upon itself. The postmodern fascination with self-reflexivity can be attributed to the sense of pastness that permeates turn-of-the-century, or rather, turn-of-the millennium culture – a sense that will endure until the new millennium finds its own cultural identity. But self-reflexivity could also be a response to the curiosity aroused by the development of a new medium. A case in point is *Don Quixote*, the foundational novel of modern Western literature, whose subject matter is the danger of reading too many novels. Janet Murray (1997: 97) ascribes Don Quixote's madness to the newly introduced practice of silent reading, which is itself a consequence of the invention of print and of the ensuing spread of the book.

The insecurity of both old cultures and young media regarding their purpose or direction explains why self-reflexivity is such a prominent feature of digital texts. As part of an old culture, digital texts are caught in the postmodern *episteme*, and they participate in the ideological, political and aesthetic preoccupations of their time; while as part of a new medium, they are still unsure of their contribution to art, to thought, and to culture. Both factors lead to a quest for identity that takes the form of a playful interrogation of the technology that supports them. In the present essay I will focus on the patterns of self-reflexivity found in Web-based art (or net.art), arguably the form of new media that has pursued the scrutiny of its technological foundation the most persistently. To prepare, theoretically, the ground for this investigation, I will start by offering an overview of the various forms of self-reflexivity.

1. Types of self-reflexivity

The term "self-reflexivity" covers a wide range of phenomena diversified along three continuums: the continuum of explicitness, the continuum of scope, and the continuum of individuation. The continuum of explicitness runs from a strong pole of literal self-reference through an intermediary zone of self-reflexivity to a weak pole of artistic self-awareness. Literal self-reference is illustrated by the famous paradox-creating sentence "this sentence is false". Genuine self-reference, as opposed to mere self-reflexivity, is a feature limited to semiotic systems capable of making propositions or issuing commands. Outside natural language, we find it in mathematics, for instance in Gödel's proof of the incompleteness of axiomatic systems, and in computer code, such as the recursive function that computes the Fibonacci number series by launching multiple copies of itself. Images cannot literally refer, since they lack the indexical power of language, but they can represent themselves through recursive self-embedding. The closest we find to self-reference in the visual domain are consequently pictures that contain copies of themselves, as in the heraldry figure known as *mise en abyme*, or on the box of the Laughing Cow brand of cheese, where we see a cow with earrings representing the Laughing Cow box of cheese. The middle of the spectrum of explicitness is occupied by works that present what I will call symbolic or emblematic forms of self-representation. Whereas the type of straight self-reference that we find in "this sentence is false" represents nothing outside itself, symbolic or emblematic self-reflexivity represents both the text of which it is a part, and something situated in the world created or described by the text. In a narrative text, for instance, the description of an object or a conversation between characters may both play a role within the plot, and tell us how the text should be read, and in a poem, a metaphor may both participate in the concrete thematics of the text, and offer an image of poetry. At the weak pole of the continuum of explicitness we find the self-awareness that Roman Jakobson (1960) calls the "poetic function of language". Jakobson divides acts of communication into six parameters (the sender, the receiver, the message, the context, the code and the physical channel that puts the sender in contact with the receiver), and he associates each of these parameters with a specific function. Among these, the poetic function is the one that focuses on the message for its own sake. (Here message must presumably be understood as an inseparable union of form and content.) Verbal art, in other words, is language that attracts attention to itself, but it can do so in a subtle way, through either pleasant sound patterns or creative imagery, without explicitly taking itself as referent.

The continuum of scope diversifies self-reflexivity according to how much of the text the self-reflexive elements capture in their mirror, and how dominant

they are in the global economy of the text. Linguistic self-reference, as we find in "this sentence is false", illustrates perfect scope, since the range of the indexical element "this" encompasses the entire sentence, and since the sentence does nothing else than reflect upon itself. Visual self-reference, by contrast, is always incomplete. An image can only show the image in which it is embedded if it also shows itself as part of the larger image; but for this copy of itself to be faithful, it must contain a third copy, and so on in an infinite regression. Whereas the Laughing Cow box illustrates explicit but incomplete self-reflexivity, the opposite situation is represented by an emblematic text that models itself entirely through symbolism, as is arguably the case with Mallarmé's hermetic poems about poetry. A text may be self-reflexive throughout – in which case it becomes an allegory of itself – or blend reflexive and non-reflexive elements. The scattered reflexive elements may furthermore represent particular aspects of the text, rather than trying to mirror it in its totality.

The third continuum concerns the focus of the reflexive activity. It runs from texts that reflect specifically on themselves, highlighting their distinctive features, to texts that include a broader class in their self-mirroring, such as their medium or their genre. I will call these two poles individuated and categorial self-reflexivity. As an example of categorial self-reflexivity, consider the lexia "This writing" from the hypertext *Patchwork Girl*, by Shelley Jackson (1995), which reflects on the difference between reading from a book and reading on a screen in a hypertext environment:

> When I open a book I know where I am, which is restful. My reading is spatial and even volumetric. I tell myself, I am a third of the way down a rectangular solid, I am a quarter of the way down the page, I am here on the page, here on this line, here, here, here. But where am I now [reading hypertext]? I am in a here and a present moment that has no history and no expectations for the future.
>
> Or rather, history is only a haphazard hopscotch through other present moments. How I got from one to the other is unclear. Though I could list my past moments, they would remain discrete (and recombinant in potential if not in fact), hence without shape, without end, without story. Or with as many stories as I care to put together.

While these remarks outline a theory of hypertext that purports to describe the medium itself, rather than one of its particular instantiations, *Patchwork Girl* also includes self-reflexive elements that distinguish it from other works of hypertext fiction: for instance, the text map for the section "Crazy Quilt" is deliberately shaped like a patchwork quilt. This image alludes to the narrative thematics of the text, which describes how a fictional counterpart of Mary Shelley creates a female monster by sewing together the body parts of various women. The sewing activity of Mary Shelley functions in turn as an allegory of the writing activity of the author, Shelley Jackson, who stitches together the body of a text out of heterogeneous (and often recycled) textual fragments. In a movement leading from individuated to categorial self-reflexivity, the shape of the map allegorizes the particular story, and the story allegorizes the type of writing promoted by the Storyspace authoring system, with which *Patchwork Girl* was composed.

2. Net.art

By net.art, I mean any artwork available for free on the World Wide Web that takes advantage of the computer, not only as a mean of production and dissemination, but also as a support necessary to the performance of the text. In other words, I restrict net.art to works that need to be executed by code. This definition excludes any artwork meant to be printed (such as Photoshop art or standard literary texts posted on the Web), as well as any work sold in CD form (hypertext fiction, computer games), but it accepts both works that can be run directly from the Web, and works meant to be downloaded and executed on the user's computer.

Net.art was born in the nineties, when the Internet developed from a resource mainly used by a technologically savvy elite into a widely accessible forum of mass communication, information, entertainment, and commercial activity. It represents the revenge of the hackers, who previously owned cyberspace, over the general public who now crowds (and spoils) the formerly guarded territory. Most net.art is indeed created by artists with an extensive knowledge of programming, or alternatively, by teams that include both artists and programmers. Fiercely anti-commercial – it cannot be sold to collectors and museums, hung on a wall, or placed on a bookshelf – and generally anti-utilitarian, net.art restores the old *cliché* "art for art's sake" to its full meaning. Its spirit is generally subversive, if not destructive, and its aesthetics tends to sacrifice pure beauty to conceptual interest. The vast majority of the works reproduced in Rachel Greene's book *Internet Art* (2004) give little pleasure to the eye, but the best of them stimulate the mind through the cleverness of their generative idea. While few of these works directly reflect about themselves, a very large proportion of them alludes to the features, protocols and utilities of the Internet: browsers, e-mail, and search engines. Others take shots at commercial applications, such as computer games or, as we will see, graphics programs. It could perhaps be argued that by commenting on other Internet applications, works of net.art direct reference away from themselves, and do not consequently qualify as self-reflexive. To this objection I reply that a net-supported work that takes as its subject matter a use of the Internet engages in a categorial form of self-

reflexivity, since it is itself a product of the technological environment toward which it directs attention.

In what follows, I will examine several of the original ways in which net.art comes to grips with the net: parody, codework, creative destruction, and mapping. Though I will classify my examples under one or the other of these headlines, they may participate in more than one category.

3. Parody

My example of parody does not come from an artist who specializes in net.art, but from a distinguished novelist with a predilection for technological subjects: Richard Powers's novel *Galatea 2.2* (1995) deals with artificial intelligence and the Turing test, and *Plowing the Dark* (2000) with virtual reality. Powers's web-based story "They come in a steady stream now" represents for him an incursion into a new territory. A spoof of e-mail, the story combines reflection on the technological medium with a more individuated form of self-reflexivity: the text not only takes the proliferation of spam as its subject matter, it also mimics the interface of a standard e-mail program (Figure 1).

When we first open (or rather, execute) the text we are faced with a display that looks like a mailbox with various folders: "inbox", "drafts", "sent", and "trash". As the reader clicks on a mail to read it, another message (or rather, its headline) appears on the screen. At the end of the reading process, there will be 17 mails in the inbox, but, ironically, none in the trash can, even though ten of them are spam: the user's agency is limited to reading the inbox, and in keeping with the theme of the story, the fictional system is unable to filter out the junk. The spam letters run the familiar gamut of pornography, drug offers, and investment opportunities. "Iris Suarez" peddles a catalog of singles available for dating, "Cora Triplett" advertises "http.//naughtygowild.info", "Christian Mortgages USA" tells the user "Jesus loves you – refinance now!", "Candrgs" sells 6000 medicines at "substantial price savings", "Evidence Eliminator" warns the reader that he is "in serious trouble – it's a proven fact", but offers an absolutely safe protection against this danger, and I leave the message of "Manure E. Griddlecake" to the reader's imagination. In addition to the junk mail, the mail program is plagued by pop-up messages, which readers must close one by one before opening a new mail, and each screen contains a clickable animated ad. Both of these features promote obsessively (but rather refreshingly, given its non-commercial character) the literary Web site Ninth Letter of the University of Illinois at Urbana-Champaign, where the story is posted.

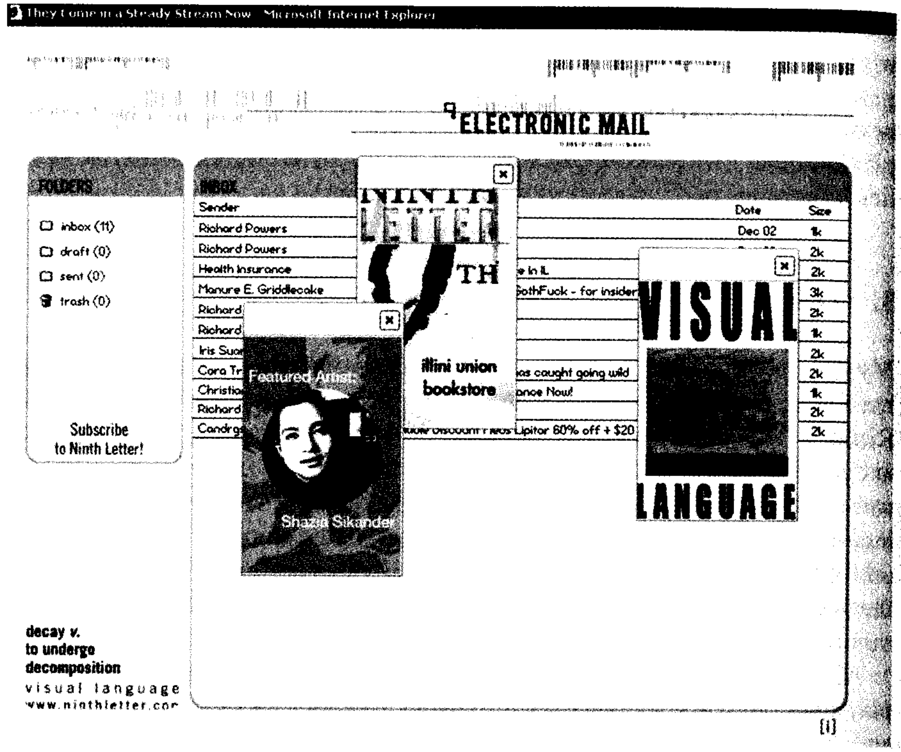


Figure 1. Screenshot from Richard Powers's "They come in a steady stream now"

Counterbalancing the humor of the junk mail, the seven "legitimate" letters, addressed to the reader by Richard Powers himself, contain a melancholic meditation on aging triggered by the spam letters' incessant hawking of drugs that promise to reverse the damage of time. The narrator sees himself on the brink of a brave new world inhabited by a posthuman species that enjoys eternal youth, constant state of sexual desire, and perfect memory, but he realizes that, like Moses, he will never enter this Promised Land:

> Lifestyle drugs, they're called: and who is going to argue? Not you, at 65, the last member of the last generation of humans still barred from returning to the garden, the last who will have to grow old, with nothing to look forward in retirement but Internet come-ons from the eternal future... What will it feel like, to be another species? Nothing that your species might compare it to. Soon we'll be whatever comes after people. And puzzled by the hunger that we've finally outgrown. (Letter 4)

While the spam outlines a dystopic future (at least for those who value our present condition of pre-posthumans), it also opens windows onto the past by jogging the memory of the narrator, not through drugs, but, quite inadvertently, though the randomly generated names of the fake senders. A mail reminds the narrator of the first girl he loved at age fifteen, and he wonders "whether she ended up as graceful as she began", but unfortunately, "her name is too common to Google". Another message – the sixth of the seven letters of the series – mentions a mail that bears the name of "a boy from your confirmation class, struck by lightening when scrambling out of a lake one summer", but all the narrator remembers about the boy is his auburn hair, his kindness, and his "goofy smile that declared a standing state of total bafflement at the passage of time".

This sentence foreshadows the reversal of time's arrow that will happen at the end of the story, but not before the reader submits to a common Internet ritual. In the last of the seven letters we read: "PLEASE REGISTER. The content you requested is available only to registered members. Registration is FREE and offers great benefits." The user is asked to enter his e-mail address in a box, and to submit it by clicking a button. At this point I hesitated, wondering what kind of plague I would bring upon my system by following these instructions, but in the end, curiosity prevailed over caution. I was rewarded with a response in the best tradition of Amazon.com: "Thank you! You will receive your confirmation e-mail shortly."

The real e-mail sent to the user consists of a link to an Adobe file that can be downloaded and then printed. This file contains the text of the previous six fake mails, together with a very Proustian conclusion. In the new segment, the narrator recaptures the lost time with a glance outside the window that liberates him from the dystopic future of the screen, sends him back to the present, refreshes his memory (without drugs!), and eventually leads to an absorption of the past by the present, allowing the narrator to relive in its full intensity the glorious day at the lake before the boy was struck by lightening. By including all the previously read installments, the final delivery invites the reader to reflect on the difference between the print and the electronic medium. The text that came to us as a collection of fragments in the e-mail simulation achieves a closure and unity in the printable file that gives rise to an entirely new reading experience. Straddling two media, the text contrasts the continually interrupted reading that takes place on the screen with the appreciation of the poetic quality of its language that becomes possible when we hold the whole story in our hands. The originality of Powers's achievement lies in the complementarity of the comic experience of the screen version and of the lyrical experience of the print version. In its play with two media, the text is truly more than the sum of its parts.

4. Codework

Codework is a reaction to the so-called WYSIWYG (what you see is what you get) aesthetics that has dominated the design of software and operating systems since the Macintosh personal computer inaugurated the graphic user interface in the early eighties. Before that time, users communicated with the machine by typing instructions on the (infamous) command line of the DOS operating system. These instructions, which had to be memorized and typed exactly, could be regarded as a high-level programming language. It took some knowledge of its functioning to operate a computer, and the difference between user and programmer was much smaller than it is today. With the introduction of the graphic interface, all the user has to do is to click on an icon to launch an application, and anything resembling coded instructions becomes invisible. For the common user, this was a blessing; for the hackers, who saw themselves as the guardians of an esoteric knowledge, this was a profanation of the machine. Icons are perfectly opaque buttons, and clicking on them requires no more knowledge of the inner working of the machine than choosing an item on the touch-operated menu of your microwave oven. Codework is an attempt to restore the user's awareness of the hidden layers of machine instructions that make it possible for data to travel from the depth of computer memory to the surface of the screen.

The play with code in net.art takes various forms. The most superficial – with regard to the deeper layers of computer architecture – is a blend of typographical signs borrowed from human and computer languages. Here are samples of the pidgin languages invented by two practitioners of this technique, Mez (pseudonym for Mary Ann Breeze), and Talan Memmott:

> Mez:
> if:
> prealphanumeric//pre network n-cluded use ov com.put
> [ty/fillah]ers offline
> then:
> n-turr-rest in nework system[ic]z stemmed fromme a more organic base,
> collaborationz via real-time fleshmeat N n-stallation based
>
> Memmott:
> From out of NO.where, Echo appears in the private space of Narcissus.tmp to form a solipstatic community (of 1, ON) with N.tmp, at the surface. The two machines – the originating and the simulative – collapse and collate to form the terminal-I. a Cell.f, or cell. . . (f) that processes the self as outside of itself—in realtime.

If the hybridization of human subjectivity and computer intelligence prophesized by the theorists of the cyborg and the posthuman (Haraway 1989; Hayles 1999) ever becomes reality, this kind of language could develop into the literary idiom of the new species. But the use of typographic elements borrowed from computer languages will remain a purely cosmetic phenomenon as long as the text cannot be run by the computer. For a school of net.artists that includes Florian Cramer, Eric Andreychek, and John Cayley, codework should not only address human concerns when read as a text, it should also change the state of the system when executed as code; otherwise we could just as well read regular code as a literary text; or feed the binary version of a literary text to the computer as executable program and watch it cause the run-time error of "unknown instruction".

Yet another form of play with code consists of revealing the actual commands that underlie a text. This was the purpose of CODeDOC, an exhibition organized in 2002 by Christiane Paul at the Whitney Museum of American Art in New York City. Paul, the adjunct curator of New Media Arts, gave a dozen artists the assignment to write a computer program whose purpose was to connect and move three points in space, a theme that could be interpreted either literally or figuratively. The exhibit inverted the usual hierarchy between code and output, by making visitors (as well as users of the Web site where the project now resides) scroll through the code file, until they reached a button at the bottom that triggered the execution of the program.

The projects vary widely in their faithfulness to the given theme, and most of them limit self-reflexivity, beyond the fact that the code is made visible before its output can be experienced, to the embedding of a description of the purpose of the program as non-executable comments within the code file. But two of the projects carried self-reflexivity beyond telling us "look, I'm made of code" by creating an individuated connection between the code and its output.

In the first of these two projects, "Jack & Jill" by John Klima, the code produces an imitation of the low-resolution computer games of the eighties, such as Lode Runner or Donkey Kong. The task of connecting and moving three points in space is ingenuously and humorously fulfilled by turning the three points into the protagonists of the well-known nursery rhyme "Jack and Jill". The purpose of the game is to enact the plot of the nursery rhyme, by taking Jack and Jill up a slope to fetch a pail and by making them tumble down the hill, once the pail has been reached (Figure 2).

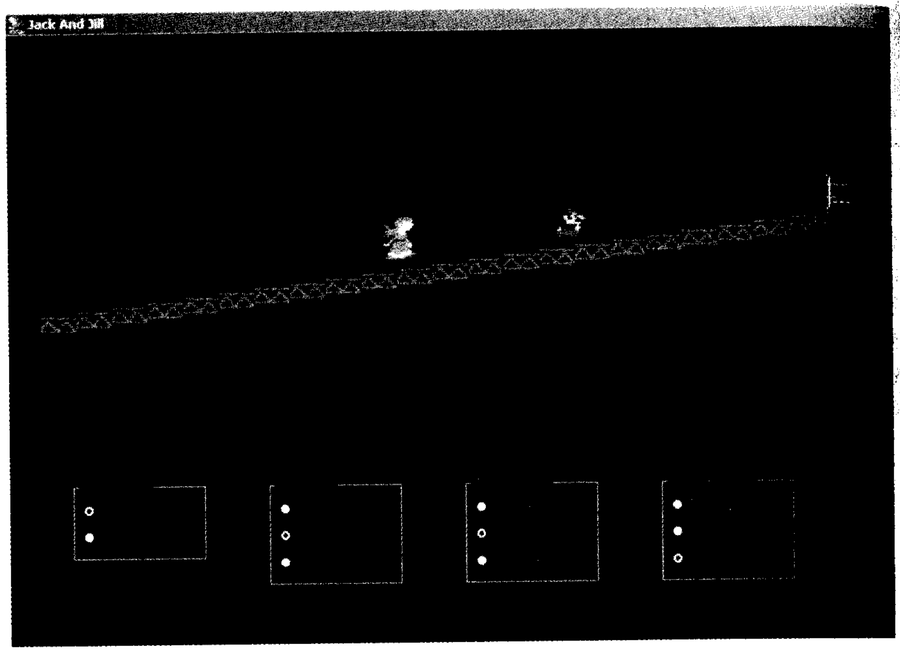


Figure 2. Screenshot from John Klima's "Jack & Jill"

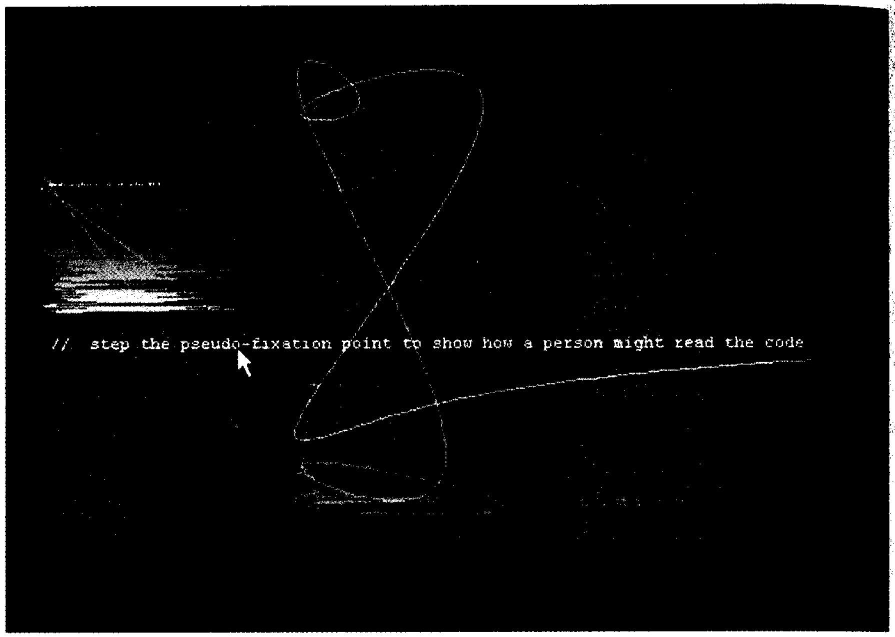
In contrast to standard computer games, the user cannot use the keyboard to control the characters, but he can influence their movements indirectly by assigning values to a number of variable parameters: the choice of a "Chauvinist" or "Feminist" attitude decides which character is ahead of the other; the assignment of an intensity value to Jack's and Jill's desire controls the speed at which the characters climb the hill (with a low desire, they never get to the pail); and the specification of "pail allure" (which gives a choice of repulsive, moderate or undeniable) dictates the magnetic force exercised by the pail. To win the game, the user must find the proper combination of values for the parameters. The game is too easy to really challenge the player, but the real programming coup lies in the duplication of the game story by the text of the code. In other words, the story is both dramatically enacted on the screen, and verbally narrated in the code. In contrast to most of the other projects of the exhibit, "Jack & Jill" makes it rewarding, not only to *look* at the code, but to actually *read* it:

```
Sub Main()
  The_Story.Show
  While True
    If YourAttitude = CHAUVINIST Then
        If Fetch(pail, jack, jill) then GoUpHill jack, jill
        If FellDown(jack) and BrokeCrown(jack) then TumblingAfter
        jill, jack
    Else YourAttitude=FEMINIST Then
        If Fetch(pail, jill, jack) then GoUpHill jill, jack
        If FellDown(jill) and BrokeCrown(jill) then TumblingAfter
        jack, jill
    End if
    The_Story.Draw
  Wend
End Sub
```

What enables digital code to tell stories (or to produce poetry) is the fact that computer languages consist of two types of elements: names and operators. While the operators are expressed through a fixed vocabulary of reserved words specific to the language, the names (which stand for variables, constants, programs and subprograms) can be freely chosen by the programmer. In the Jack & Jill example, the story is suggested by the variables Jack, Jill and Pail, as well as by the subprogram names Fetch, FellDown, BrokeCrown and TumblingAfter, but the operators If...Then are detrimental to narrative meaning, because a story is a report of facts, and as such, it cannot be told, at least not literally, in the conditional mode (even less through embedded conditionals). The only operator that contributes to the narrative reading is =, which can be read as the verb "to be". It would be an extraordinary achievement to enroll both names and operators in the production of a story, and Klima can be forgiven for not achieving what is probably an impossible feat.

While in "Jack & Jill" the code mirrors the story told in the output, Brad Paley's "Codeprofiles" performs the reverse operation: here the output of the program is an image of its own code. Not only does the program display a listing of itself, it also fulfills the requirement set by the organizers of the exhibit by moving three points across the display according to a logic described in a comment section of the code file:

```
// This code reads in its own source and displays it in a tiny font, then //
// It moves three points in "code space." It essentially comments on itself.//
// The white Insertion Point traces the code in the order it was written. //
// The amber Fixation Point traces word by word as someone might read it. //
// The green Execution Point shows a sample of how the computer reads it. //
// The code lines themselves gradually get brighter as they execute more. //
```



```
// step the pseudo-fixation point to show how a person might read the code
```

Figure 3. Screenshot from W. Bradford Paley's "Codeprofiles" (detail)

Figure 3 shows a portion of the screen (three columns out of four). The amber point corresponds to the bright area in the left column. It runs linearly from the first to the last line, and then returns to the top, simulating the reading of a standard print text. If the user moves the cursor on one of the lines, it is magnified and made legible; if the user clicks, the execution restarts from there. The trajectory of the white point corresponds to the curved line that runs all over the image; at the moment shown in Figure 3, the point is highlighting text in the third column. Writing code is always a relatively linear process, because programmers must simulate in their mind the operation of the computer, which takes and executes the instructions sequentially, but a well-structured computer program consists of various self-contained modules, known as procedures or subroutines, which can be written in any order. This freedom explains the capricious arabesques of the white line. The movements of the green point trace the order of execution, whose sequentiality is frequently broken by commands implementing transfers of control, such as go-to statements and calls to subroutines that make the program jump across computer memory, where the instructions are stored before being brought to the processor to be executed. When I captured the program, execution followed a loop represented by the triangle between the first and second columns.

Though "Codeprofiles" takes self-reflexivity further than any of the other projects of the CODeDOC exhibit, the author claims in a discussion of the program available on the exhibit's Web site that it was not written "to be computer-clever, nor postmodern reflexive", but to compare and contrast three modes of parsing: the layman's, who is tempted to read the file linearly, like an ordinary text; the programmer's, who composes the code module by module in a relatively free order; and the computer's, whose order of execution bounces back and forth between modules and travels "code space" in all directions.

5. Creative destruction

There is perhaps no better way to make people appreciate what they have – or rather, what they *had* – than to take it away. Alan Liu (2004) suggests the term of "creative destruction" for the application of this principle in art. A practice that originated in Dadaism and Surrealism but exploded in new media, especially in net.art, creative destruction draws attention to cultural, commercial and technological phenomena by taking them apart.

In *Auto-illustrator*, Adrian Ward combines the idea of creative destruction with parody and reflection on code into a humorous piece of dysfunctional software. *Auto-illustrator* (Figure 4) mimics graphic programs, such as Photoshop or Corel PhotoPaint in the same way Richard Powers's text mimics e-mail, but with the significant difference that the interface is actually operative: you can produce your own artwork by using the program, and you can even buy a licensed copy, which contains more features than the free demo version available on the Internet. The main reason for buying a license is to support the cause of net.art, for I cannot imagine that anybody would have sufficient need for *Auto-illustrator* to pay to $100 for it. But don't expect to enjoy the program for a long time if you don't buy the license: every time you run your free copy, its performance deteriorates, until you become unable to do anything with it.

Auto-illustrator subverts the utilitarian spirit of commercial software by turning the graphic tools into autonomous agents with a will of their own. If you select the freehand pencil tool, the system does not use the position of the mouse cursor to draw a line, but rather follows its own rules, merely "taking clues from your mouse coordinates". The exact nature of these clues remains a mystery: the line you draw stubbornly refuses to follow the line you wanted to draw. If you select the text tool, the system picks the letters, inventing nonsense words, and your control is limited to making a selection among the options "terse",

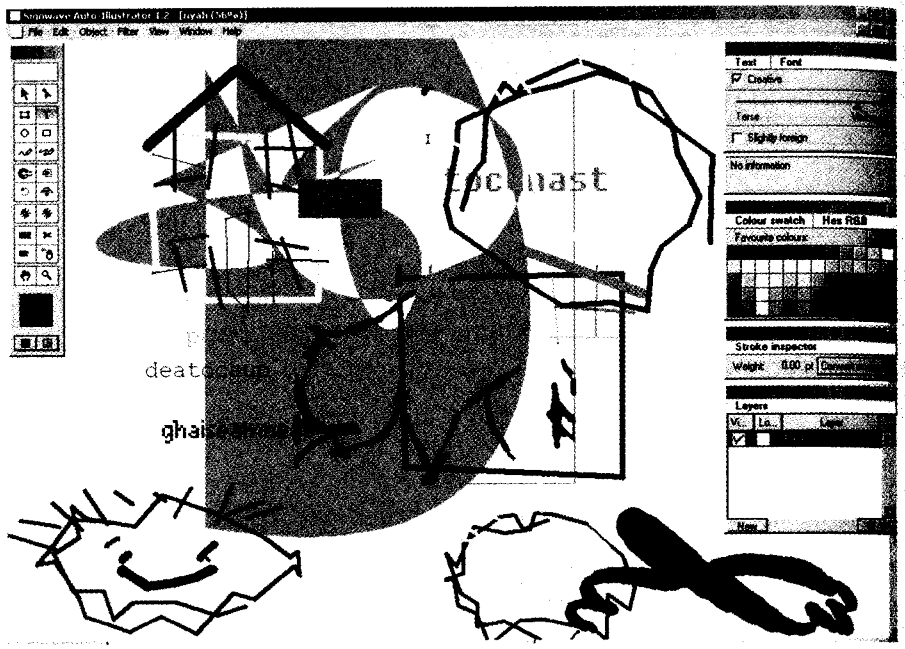


Figure 4. Screenshot from Adrian Ward's Auto-illustrator 1.2

"verbose", "creative", and "slightly foreign". The square and the oval tools let you draw regular geometric shapes, but it gives you a choice between "shabby" and "precise" shapes, as well as between "childish", "artistic", and "regular". "Artistic" does not draw anything – there is no such thing as an artistic square or circle, according to the program – but "childish" brings delightful surprises: the circles will be funny faces, and the squares will be turned into the kind of houses that a four-year-old may draw (especially if you combine the childish and shabby options). As for the bug tool, it will place moving creatures randomly on your screen, and they will create art for you by crawling around and drawing lines. If you don't like the result, a tool will let you exterminate the creatures. The parody of serious art programs extends to the system's comments on the choices of the user ("this tool is boring"), and to the zany options offered on the "preferences" menu: here the user can click boxes labeled "Death penalty for poor designs", "Exta-verbose KJX routines", "Do cool things". Her curiosity will be tested with a Pandora box labeled "don't push this button". If she succumbs to the temptation, the program's behavior will become totally erratic, but fortunately, licensed users can undo the damage by hitting a certain key on the next run of the program.

The rebellious behavior of the tools of *Auto-illustrator* reminds the user of the additional level of mediation that distinguishes drawing on paper with hand and pencil from drawing on a screen with computer software. In a graphics program, the hand does not draw, but rather activates a hidden code. The user of commercial, utilitarian software takes it for granted that the code listens to her input: if she selects the straight line tool, she does not expect the program to draw an arabesque. *Auto-illustrator* breaks this basic contract between the software designer and the user, and draws attention to the hidden code by complicating (rather than severing) the relationship between the movements of the hand and the behavior of the tools. The program does listen to the user, but it does so in an indirect, unpredictable way. This "disturbingly lively" machine – to paraphrase a much quoted formula by Donna Haraway (1989: 176) – does not produce a "frightfully inert" user, to conclude the formula, but on the contrary, distributes authorship among three agents: the programmer, who designs the code and invents imaginative new tools, the computer, whose unpredictable operation is regulated by random numbers invoked by the code, and the user, who retains modest control over the picture by choosing tools and colors, by letting the program duplicate or animate objects, and by deciding when the output is worth saving as an artwork.

Auto-illustrator's reflection on code does not take the form of making it directly visible, but rather, of asserting the artistic dimension of the programmer's activity. In other words, it is not codework, but rather, what Christiane Paul calls "software art" (2003: 124). In an article included in the user's guide to *Auto-illustrator*, Florian Cramer observes that in commercial applications, "programmers are frequently considered to be mere factota, coding slaves who execute other artist's concepts" (2002: 102). Software art liberates programmers from the tyranny of corporate work by letting them express their own vision, using code as a meta-medium to control other media: language, sound, color, shapes, and animation.

6. Mapping

The development of maps of cyberspace – by this I mean visual representations of the information contained in the Internet – is an area of teeming activity, both in net.art and in practical programming. The mapping projects inspired by the Internet (many of which are shown in Dodge and Kitchin's fascinating *Atlas of Cyberspace* [2001]) range from purely functional navigational tools through projects that combine usefulness and artistic self-awareness to artworks totally devoid of practical purpose. Here I will discuss an attempt to map the Internet aimed at resolving a paradox that has fascinated authors as illustrious as Lewis Carroll and Jorge Luis Borges: the paradox of the map that achieves perfect self-referentiality by becoming indistinguishable from the represented territory.

A fusion of map and territory would necessitate a complete image of a territory at a 1 to 1 scale that includes the map itself. Why 1 to 1? Because any reduction would require the omission of some features. And why should the map be part of the territory? Because if it weren't, it would point to something external to itself: one can for instance imagine a complete map of the earth at a 1 to 1 scale spread out on another, larger planet. Both of these conditions lead to paradoxes. As Borges has shown ([1951] 1983: 195–196), if a map is part of the world, it can only represent the world completely by representing itself, which means that it must represent its own self-representation, in an infinite regression similar to the case of the Laughing Cow box of cheese. Moreover, if the map were at a 1 to 1 scale, it would cover the whole world, and according to Lewis Carroll this would lead to inevitable contradiction. A perfect map should contain an image of every blade of grass, but if it were spread out over the world, the sun would be blocked, the grass would die, the farmers would be mad, and the map would be unfaithful. And if the map were not spread out... it could not be consulted, and it would become useless. Carroll suggests, tongue in cheek, a luminously simple solution to this problem: "So we now use the country itself, as its own map, and I assure you it does nearly as well" ([1893] 1982: 726.). But if we think of maps as navigational aides, this is a ludicrous proposal, because we would have to traverse the territory to see what its map looks like, when in fact the purpose of maps it to help us find our way in the territory.

The digital artist Lisa Jevbratt proposes to reconcile functionality and exhaustive coverage of the territory with a mapping of the Internet appropriately titled *1:1*. According to comments by Jan Ekenberg posted on the project's Web site, *1:1* "becomes not only the map, but the environment itself". Referring to Lewis Carroll, whose text is quoted in his commentary, Ekenberg concludes: "Let's hope the farmers don't object." Jevbratt's own on-line description of the project concurs with Ekenberg's assessment: "The interfaces/visualizations are not maps of the Web, but are, in some sense, the Web. They are super-realistic and yet function in ways images could not function in any other environment or time."

The project that inspires such hyperbolic statements is an attempt to visualize the Web as a system of IP addresses. The IP address of a Web site is the numeric translation of its domain name; in other words, what is for human users www.selfreflexivity.org could be for the computer 217.170.37.221. Since IP addresses are made of four eight-bit words, for a total of 32 bits (or at least were made in 1999 and 2001, when *1:1* was created), there could be as many as 2^{32}

distinct pages on the Web; but many IP addresses are not claimed, and attempts to reach them leads to the message: "cannot find server, or DNS error". Other addresses are claimed, but the user is not authorized to access them.

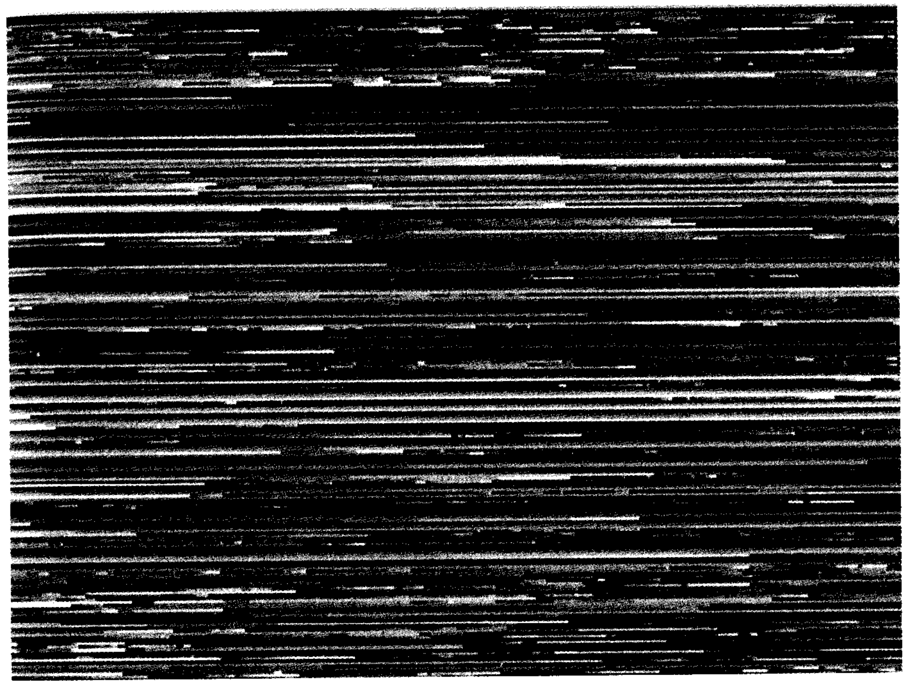


Figure 5. Screenshot from Lisa Jevbratt's *1:1* (detail)

The *1:1* project consists of five different visualizations, but I will limit my discussion to "Every", the design that makes the strongest claim of being the Web itself (Figure 5). To produce her images, Jevbratt used Web crawlers – programs that search the Web address by address – to determine which IP numbers have active servers. The crawlers returned 186 100 active addresses for the sampled areas, and each of these addresses is represented on the screen by a distinct pixel. The pixels are color-coded on the basis of the numerical value of the address they represent, so that, by looking at the image, one can tell the "distance" (in numerical value) between occupied addresses: sharp contrasts in color mean that there are large intervals between active IPs, gradual contrast means that a region is densely populated. Each pixel is a hot link, and by clicking on it the user can reach the corresponding IP. This provides an interface to the Web radically different from the modes of navigation offered by standard browsers. As Jevbratt explains on the project's Web site, "Instead of advertisement, pornography, and pictures of people's pets, this Web is an abundance of inaccessible information, undeveloped sites and cryptic messages intended for someone else". The user gets an idea of how small the proportion of the information stored on the Web is publicly accessible, and of how much the Web has changed since the creation of the project. I clicked about 20 times on the visualization, and my random selection yielded only one accessible web site: the home page of "Marjorie Orr, top international astrologer".

How should we understand the title *1:1*? One obvious interpretation is that each unit on the screen corresponds to a distinct IP address, in a one to one relation. But this relation is very different from the scale of a map, where 1:1 means that a certain area of the map corresponds to the same area in the world. The units on the screen are made of one pixel, but they stand for addresses made of 32 bits. Nor can we interpret *1:1* as meaning that the design represents the information available on the Net in its totality. The image on the screen admittedly provides access to every active IP address, but we have to traverse the image to see the content of these addresses, which means that we cannot see all of this information in one glance, as a map would let us do. Nor does the visualization show what makes the Web a web: the complex system of links that interconnects its various elements.

All this should make it clear that, while *1:1* could in principle be extended to cover the entire address field of the Web, it remains a long way from achieving the self-referentiality inherent to the claim that it is not a map of the Web, but rather the Web itself. By subjecting Jevbratt's comments to a critical assessment, rather than accepting them at face value (as most commentators seem to do, with the exception of George Dillon), we learn that the tendency of conceptual art to produce auto-descriptions does not guarantee the validity of these descriptions. A representation of an artwork is liable to be considered inaccurate, whether it is contained in the artwork itself, or describes an external referent. But if *1:1* does not really fuse the map and the territory, it remains an impressive achievement in data visualization, not only because it reveals the hidden geography of IP addresses – in this sense it is truly a map – but also because its combination of representation and active interface to Web sites creates a type of image that could only exist in a digital environment.

7. Conclusion

Let me return, in conclusion, to the question of what makes self-reflexivity so dominant in net.art. I believe that we cannot achieve a proper understanding of self-reflexivity in art in general, and in new media in particular, without

taking into account the force that it is trying to resist, namely the immersive power of representations and their ability to create an illusion of reality (Ryan 2001; Wolf 2004). The self-reflexivity of *Don Quixote* was a warning against the tendency of readers to immerse themselves in the world of chivalric novels, and to mistake these fictional worlds for reality. In the nineteenth century, the development of the powerful illusionist techniques of realism led the novel away from self-reflexivity, and steered it back toward immersion, until postmodernism denounced any attempt to make the medium invisible (a prerequisite to immersion) as robbing the reader of his critical faculties. For those who regard immersion as a low-brow pleasure (unjustly in my view, for the experience requires a highly active involvement of the imagination), replacing transparent windows into imaginary worlds with the mirrors of self-reflexivity is a proven key to artistic respectability. It is indeed by developing self-reflexive features that computer games, a fundamentally immersive use of digital technology, have recently tried to promote themselves as an art form to be taken seriously.

In contrast to the novel and to computer games, net.art never developed immersive features; what it is trying to undermine is not its own power to create illusion, but rather the kind of immersion in digital technology that limits our attention to the surface of the computer screen, and fools us into believing that we fully control this technology, when in fact our agency is restricted to what the system was programmed to let us do. As part of this attempt to provoke reflection on the role of digital technology in our lives, net.art fills the World Wide Web with images and inverted images of its own utilities. By inspiring, enabling, and hosting these multiple and varied images, the Web as a whole becomes a system that thinks about itself. Do not expect net.art to grow into an immersive art form any time soon: there are already enough of these in the media landscape. For net.art, reflecting on its supporting medium is not a search for identity, it *is* identity.

References

Borges, Jorge Luis
[1952] 1983 Partial magic in the Quixote. In: Donald A. Yates and James E. Irby (eds.), *Labyrinths*, 193–96. New York: Modern Library.

Carroll, Lewis
[1893] 1982 *Sylvie and Bruno. The Complete Illustrated Works of Lewis Carroll*. Ed. Edward Guillaro. New York: Avenel Books.

CODeDOC exhibit.
2002. http://artport.whitney.org/exhibitions/index.shtml (08.02.06).

Cramer, Florian
2002 Concepts, notations, software art. *Auto-illustrator Users Guide*: 101–112. Downloadable from http://www.Auto-illustrator.com/ (08.02.06).

Dillon, George
2002, 2003 .*Writing with Images: Toward a Visual Semiotics of the Web*, http://courses.washington.edu/hypertxt/cgi-bin/12.228.185.206/html/ (08.02.06).

Dodge, Martin and Rob Kitchin
2001 *Atlas of Cyberspace*. New York: Addison-Wesley.

Ekenberg, Jan
2001 Prologue to 1:1, http://128.111.69.4/~jevbratt/1_to_1/jan.html (08.02.06).

Greene, Rachel
2004 *Internet Art*. London: Thames & Hudson.

Haraway, Donna
1991 *Simians, Cyborgs, and Woman: The Reinvention of Nature*. London: Routledge.

Hayles, N. Katherine
1999 *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics*. Chicago: University of Chicago Press.

Jackson, Shelley
1995 *Patchwork Girl*. Hypertext software. Cambridge, MA: Eastgate Systems.

Jakobson, Roman
1960 Closing statements: Linguistics and poetics. In: Thomas A. Sebeok (ed.), *Style in Language*, 350–77. Cambridge: MIT Press.

Jevbratt, Lisa
2001 1:1, http://128.111.69.4/~jevbratt/1_to_1/index_ng.html (08.02.06).

Klima, John
2002 *Jack & Jill*, http://artport.whitney.org/commissions/codedoc/klima.shtml (08.02.06).

Liu, Alan
2004 *The Laws of Cool: Knowledge Work and the Culture of Information*. Chicago: University of Chicago Press.

Memmott, Talan
2000 *Lexia to Perplexia*, http://www.uiowa.edu/~iareview/tirweb/hypermedia/talan_memmott/ (08.02.06).

Mez. [Mary Anne Breeze]
2000 *The Art of M[ez]an.ell.ing: constructing polysemic & neology fic/factions online*, http://beehive.temporalimage.com/archive/34arc.html (08.02.06).

Murray, Janet
1997 *Hamlet on the Holodeck: the Future of Narrative in Cyberspace*. New York: Free Press.

Paley, W. Bradford
2002 *Codeprofiles*, http://artport.whitney.org/commissions/codedoc/paley.shtml (08.02.06).

Paul, Christiane
2003 *Digital Art*. London: Thames and Hudson.

Powers, Richard, Jenifer Gunji, Joseph Squier, Jessica Mullen, Lauren Hoopes, Chad Kellenberger and Val Lohmann
2004 *They Come in a Steady Stream Now*, http://ninthletter.art.uiuc.edu/FA/FA05/ (08.02.06).

Ryan, Marie-Laure
2001 *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*. Baltimore: The Johns Hopkins University Press.

Ward, Adrian
2003 *Auto-illustrator 1. 2*. Downloadable from Signwave: http://www.auto-illustrator.com/ (08.02.06).

Wolf, Werner
2004 Aesthetic illusion as an effect of fiction. *Style* 38(3): 325–51.