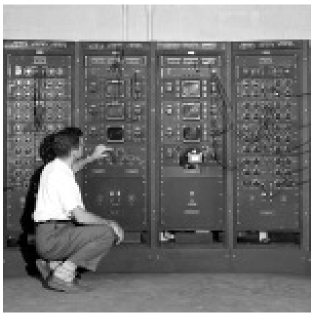
The Code is not the Text (Unless It Is the Text)

electronicbookreview.com/essay/the-code-is-not-the-text-unless-it-is-the-text

John Cayley
by John Cayley
09-10-2002

thread: electropoetics
This essay was peer-reviewed.
electropoetics



An argument against the collapse of categories by an author who has, yes, himself perpetrated a few codeworks.

Digital utopianism is still with us. It is with us despite having been tempered by network logistics and an all-too-reasonable demand for 'content.' Admittedly, New Media has aged. It has acquired a history or at least some genuine engagement with the reality principle, now that the Net is accepted as a material and cultural given of the developed world, now that the dot.coms have crashed, now that unsolicited marketing email and commercialism dominates network traffic. Nonetheless, artistic practice in digital media is still often driven by youthful, escapist, utopian enthusiasms. Net Art as such pretends to leapfrog this naivety through the wholesale importation of informed, ironic, postmodern conceptualism, offering us the shock of the virtual-visceral banal at every possible juncture. Other, more traditionally delineated arts - literature, music, photography, fine art, architecture, graphics, etc. - struggle to cope with the reconfiguration of their media, or with a migration to complex new media which are suddenly shared, suddenly intercommunicable with those of artistic practices previously considered to be distinct. One way of coping is escape.

I write as a literary artist, my ever-provisional, traditionally delineated subject position in this context: poet. When asked, in social contexts, I don't really know what to call myself, although - when I manage to remember the phrase - 'literal artist' seems about right. I write as a practitioner, but I am interested in the theory underlying my practice because I recognise that my artistic media are being reconfigured to a degree which may well be catastrophic, or, at least, allow me and my fellow writers to recall that these media - textual media - have always been subject to reconfiguration. Serious formalism in literature was never just a matter of rhetorical flourish; it was inevitably, ineluctably, concerned with the materiality of language, and therefore with the affect and significance of language as such.

If you persist, you are about to read a theoretically-inflected critique of what some people call 'codework.' Potentially codework is a term for literature which uses, addresses, and incorporates code: as underlying language-animating or language-generating programming, as a special type of language in itself, or as an intrinsic part of the new surface language or 'interface text,' as I call it, of writing in networked and programmable media. Why do many of the current instantiations of codework, along with some of the theoretical writing that underpins this practice, require critique at this time? What is at stake? I have to try and briefly answer this question at the outset, because what follows is largely critical, something I wrote and felt I'd completed in response to questions which are only now being formulated; it is part of an emergent debate about the role of code in literal art. There will be much more to write, at other times and places, which is less critical and more generative, precisely because of what is at stake.

In utopia, because you are nowhere you are everywhere at once. Transparency and translatability are key values of digital utopianism. We should perhaps remain sceptical not only concerning the no-place itself but also concerning its values. Are they indeed values? In much current codework language is (presented as) code and code is (presented as) language. The utopia of codework recognises that the symbols of the surface language (what you read from your screen) are the 'same' (and are all ultimately reducible to digital encoding) as the symbols of the programming languages which store, manipulate, and display the text you read. The mutual transparency and translatability of code and language becomes a utopian value, and when it is recast as the postmodern virtual-visceral banal - as the mutual infection-contamination of language by code and code by language - it becomes a subversive (i.e. potentially progressive) utopian value. Basically, my argument in what follows is that, in much existing codework, this is as far as we get. A simple point based on digital transparency and translatability is being made in a context which is already utopian and this more or less exhausts the significance and affect of the work. If, furthermore, your focus remains fixed on the interface text - on what can be read and recorded from the screen as writing - then much critical energy goes into interpreting work with an all-but-exhausted aesthetic program in a fairly traditional and conservative manner. The code is in the text or the text is in the code, and it's there because it can be, and that's what we have to say about it.

So what is left outside of this utopia? It is obviously a tactical exaggeration to say that most instances of codework in networked and programmable media are exhausted by the aesthetic I have briefly introduced and caricatured. More accurately, there is a problem with the way code-as-text is appreciated and appropriated within the broader critical 'language of new media.' Much work exists that can not or should not be assimilated into the utopia of code-language transparency. I argue that certain reasons why such work is alien to the utopia of transparency are also precisely reasons why it is able to generate significance and affect - because the code is not necessarily transparent or visible in human-readable language; because code has its own structures, vocabularies and syntaxes; because it functions, typically, without being observed, perhaps even as a representative of secret workings, interiority, hidden process; because there are divisions and distinctions between what the code is and does, and what the language of the interface text is and does, and so on. A specialised appreciation for code does not in any way preclude the mutual contamination of code and natural language in the texts that we read on screen, it simply acknowledges that - in their proper places, where they function - code and language require distinct strategies of reading. The necessity to maintain these distinct strategies as such should lead, eventually, to better critical understanding of more complex ways to read and write which are commensurate with the practices of literal art in programmable media.

To conclude these introductory remarks, here is a suggestive and non-exhaustive list of things I believe are at stake, a list of approaches to work which risk being ignored or downgraded if we remain focused on codework as code-language transparency:

- If a codework text, however mutually contaminated, is read primarily as the language displayed on a screen then its address is simplified. It is addressed to a human reader who is implicitly asked to assimilate the code as part of natural language. This reading simplifies the intrinsically complex address of writing in programmable media. At the very least, for example, composed code is addressed to a processor, perhaps also addressed to specific human readers (those who are able to 'crack' or 'hack' it); while the text on the screen is simultaneously? asynchronously? addressed to human readers generally. Complexities of address should not be bracketed within a would-be creolized language of the new media utopia.

- Address to other, unusual reading processes - the machine itself, or particular human readers who have learned how systems read - implies the need for different persuasive strategies, different strategies for generating significance and affect. I mean that the rhetoric of writing in code must be distinct. Again, appeal to values of hybridity and mutual linguistic contamination (addressed to postmodern humans) threatens to conceal the emergence of new or less familiar rhetorical strategies. In what follows I briefly mention two of these, the tropes of strict logical process and another I identify with compilation in the programmer's sense. There is a lot of very necessary work to be done here, identifying the unacknowledged tropes and figures of literal art in new media. Perhaps even certain questions concerning the rhetoric of electronic games (when viewed as literal art) could be studied in this context. For example, the trope of 'playability' emerges as much from the composition of code as from the 'writing' (in the scriptwriter's sense) during games development.

- Reading codework as code-in-language and language-in-code also risks stunning the resultant literary object, leaving it reduced to simple text-to-be-read, whereas there are real questions of how such work is to be grasped as an object: is it text, process, performance, instrument? If code is treated distinctly, as an aspect of writing with its own structures and effects, then we gain in the potential to articulate more appropriate classes of literal objects, with instrument, for example, forming one class I would prefer, personally, to instantiate and explore.

- A question I do begin to engage in what follows is the materiality of language and how this may be evolving in writing for programmable media. I query N. Katherine Hayles' position in the code-as-text debate, particularly her readings of the work of certain codework artists along with her invocation of the 'flickering signifier,' which I suspect her of using to underpin this codework despite the fact that such work does not necessarily engage with the materiality of a flickering signifier. By the time we get to read code-as-text, in most cases it is presented as, at best, a chain of resolved floating signifiers, with the code elements simply providing a layer of associative complexity or slippage. Hayles' signifier has far greater potential and this not always operating in the code-as-text variety of literal art. The flickering signifier cannot simply be seen as something which goes on behind the screen; it emerges when code is allowed, as I say, its proper place and function: when the composed code runs. As it runs, the code is not the text, it is not a set of (non-sequential) links in a chain of signifiers; the code is what makes them flicker, what transforms them from writing as record of static or floating simultaneities into writing as the presentation of atoms of signification which are themselves time-based (they are not what they are without their flickering transformations over time, however fleeting these may be).

- The implicit requirement - at one and the same time - to pay close and particular attention to the role of code in literal art, while, at certain moments of reading, to allow that this distinct role functions in concealment, will have practical as well as theoretical effects on artists' creative methodology even if only to help them to better understand how and why they are working with code. The emergent materiality of the signifier - flickering, time-based - creates a new relationship between media and content. Programming the signifier itself, as it were, brings transactive mediation to the scene of writing at the very moment of meaning creation. Mediation can no longer be characterised as subsidiary or peripheral; it becomes text rather than paratext. Criticism of code-making in this context becomes even more important and central than, for example, the criticism of instrumentation or interpretation in musical recital. What I say about new media literary objects being reconceived as 'instruments' is not meant to imply that they are, in any sense, merely instrumental.

The question of the materiality of the signifier, in particular, is a big one, which I believe Hayles is currently readdressing and which I hope to take up in a sequel to what finally, now follows.

*

The use of networked and programmable systems as both delivery and compositional media for literal and verbal art (and other forms of new media art) has provoked critical engagements which pretend to reveal and exam the various levels of code and encoding which are constituent of programmatological systems. *Certain terms in this essay may require explanation. I prefer, despite its awkwardness and length, 'writing in networked and programmable media' to any of the current words or phrases such as 'hypertext, hyperfiction, hyperpoetry,' etc. or the corresponding 'cyber-' terms, although I do generally subscribe to Espen Aarseth's 'textonomy,' and would prefer cybertext to hypertext as the more inclusive, 'catholic' term. Espen Aarseth, Cybertext: Perspectives on Ergodic Literature (Baltimore and London: John Hopkins University Press, 1997). I use 'programmatology' and 'programmatological' by extension from* *'grammatology' and especially 'applied grammatology' as elaborated by Gregory Ulmer. Gregory L. Ulmer, Applied Grammatology: Post(E)-Pedagogy from Jacques Derrida to Joseph Beuys (Baltimore: John Hopkins University Press, 1985). Programmatology may be thought of as the study and practice of writing (Derridean sense) with an explicit awareness of its relation to 'programming' or prior writing in anticipation of performance (including the performance of reading). I try to avoid the use of the word 'computer' etc. and prefer, wherever possible, 'programmaton' for the programmable systems which we use to compose and deliver 'new media.'* The title of the section of the p0es1s programme which provoked this paper - 'Code as Text as Literature' - is a case in point. *This essay was originally sketched out for the "p0es1s: poetics of digital text" symposion (sic), held in Erfurt, 28-29 September, 2001 (http://www.p0es1s.com).* In more extreme forms of such engagement, a radical post-human reductionism may be proposed, such as that, for example, which can be read from certain of Friedrich Kittler's essays, in which the ramifications of "so-called human" culture, especially as played out on new media, become qualitatively indistinguishable from "signifiers of voltage difference" ("There Is No Software" 150), demonstrably the final, lowest-level 'ground code' of the increasingly familiar practices of cultural production which make use of programmable tools; and perhaps also essential to the brain activity which generates the objects and subjects of psychoanalysis. Kittler is reviewed by Bruce Clarke in ebr... Nowadays voltage difference accounts for and instantiates everything from the encrypted transactional play of internet banking to the promised consensual hallucination of immersive Virtual Reality. However, the purpose of this brief paper is to address a number of less productive confusions which arise from this engagement with code-as-text, citing a few examples of artistic practice and a number of critical sources. *There are times when I would like to write 'code-as-text' and other times, 'text-as-code,' occasionally with either term cycling (code-as-text-code, etc.) I will just use the one term, asking the reader to bear in mind the other possibilities in appropriate contexts.* While allowing the value of certain metacritical statements such as Kittler's (which take on questions of what culture is or may become), my aim is to disallow a wilful critical confusion of code and text, to make it harder for critics to avoid addressing one or the other by pretending that they are somehow equivalent, or that codes and texts are themselves ambiguously addressed to human readers and/or machinic processors (unless they are so addressed, however ambiguously). *As an example of the prevalence of code-as-text across the widest range of artistic inscription, a version of the code-as-text or reveal code aesthetic appears as something of a culmination in Lev Manovich's excellent and provocative The Language of New Media (not discussed in the body of the present essay because of my focus on textual and literal art practice). The final section of Manovich's book is entitled 'Cinema as Code' and features Vuk Cosic's ASCII films, "which effectively stage one characteristic of computer-based moving images - their identity as computer code." Manovich is undoubtedly correct when he asserts that, "What [George] Lucas hides, Cosic reveals. His ASCII films 'perform' the new status of media as digital data... Thus rather than erasing the image in favour of the code ... or hiding the code from us ... code and image coexist." Nonetheless, it is worrying to be*

presented, in this highlighted context, with the example of work whose aesthetic may well prove to be exhausted by a conceptual and metacritical analysis (see below), particularly in a book which makes an unprecedented contribution to our understanding of new and emergent rhetorical strategies in new media (especially the crucial role of cinematic rhetoric), and represents a deep understanding of new media's programmatological dimension. Lev Manovich, The Language of New Media, ed. Roger F. Malina, Leonardo (Cambridge: MIT Press, 2001) 330-33.

I have invoked reductionism and by this I mean a critical thrust which, implicitly or otherwise, asks questions like, 'What (ultimately) is this object we are examining? What is its structure? What are its essential or operative characteristics?' and then finds special critical significance in the answers proposed. In N. Katherine Hayles' sophisticated version of what can be read as a code-as-text argument, this reductive inclination is in evidence. Her essay 'Virtual Bodies and Flickering Signifiers' discovers a new or emergent object, the flickering signifier, and derives important consequences from its instantiations and methods. "The contemporary pressure toward dematerialization, understood as an epistemic shift toward pattern/randomness and away from presence/absence, affects human and textual bodies on two levels at once, as a change in the body (the material substrate) and a change in the message (the codes of representation)." *N. Katherine Hayles, "Virtual Bodies and Flickering Signifiers," in How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics (Chicago: University of Chicago Press, 1999), 29. An earlier version of this essay is also published as: N. Katherine Hayles, "Virtual Bodies and Flickering Signifiers," in Electronic Culture: Technology and Visual Representation, ed. Timothy Druckery (New York: Aperture, 1996).* In other words, Hayles suggests that the constituent structure of the signifier itself may be seen as changed in contemporary culture and especially as expressed in 'new media.' Both the materiality and the represented content of cultural practice and production has been affected. Before examining parts of Hayles' argument in more detail, I want simply to point out that it is clearly determined by its metacritical significance and has a reductive inclination: signifiers have come to be such and such, therefore - albeit in a cybernetic feedback loop - cultural production (in Hayles' essay "the represented worlds of contemporary fiction") follows suit. Hayles' characterization of a multiply mediated signifier which flickers from level to level in chained coded structures is, as a metacritical statement, highly suggestive and useful. However, when it comes to art practice and the critique of this practice, how does such insight figure?

What is missing from Hayles' analysis is a set of relationships - relationships constituted by artistic practice - between a newly problematized linguistic materiality and represented content. These would inevitably express themselves in formal as well as conceptual address to what she identifies as a changed matter of language and literature. Hayles' chosen examples, with, perhaps, the exception of her use of William Burroughs, demonstrate conceptual rather than formal address; they represent flickering signification as concept rather than as instantiation in the language of the work. Hayles cites, most extensively, William Gibson's Neuromancer as a prime example of represented content affected by and expressive of the flickering signifier. While Gibson brilliantly conveys the literally flickering, scanned and rasterized, apparent immateriality of an informatic realm, the 'consensual hallucination' of 'cyberspace' (his famous coinage) and its interpenetration of meatspace, he does this in a book - 'a durable material substrate' - in a more or less conventional novel, one in which, indeed, narrative predominates over character development and in which language functions in a relatively straightforward manner. Not even the narrative perspective (omniscient author third person) is shifted or experimentally inflected in any of Gibson's cyberpunk classics. The writing is sharp and inventive but entirely subject to paraphrase.

There are further significant ironies here, for Hayles begins her essay by discussing typewriting. The physicality and static impression-making of this process of inscription is contrasted with that of word processing where less substantial bodily gestures cause word-as-(flickering)-image to be scanned onto the surface of a screen. "As I work with the text-as-flickering-image, I instantiate within my body the habitual patterns of movement that make pattern and randomness more real, more relevant, and more powerful than presence and absence" (Hayles, "Virtual Bodies and Flickering Signifiers," 26). However, the exemplar most present later in her argument, Gibson, has made some play of his preference for composing his novels using a typewriter. *William Gibson (1948-) [Web site] (Guardian Unlimited, 2001 [cited February 2002]); available from http://books.guardian.co.uk/authors/author/0,5917,96528,00.html. Michael Cunningham, The Virtual Tourist [a Short Interview with William Gibson] [Web site] (P45.net, 1996 [cited February 2002]); available from http://www.p45.net/dos_prompt/columns/3.html. "In real life, Gibson is actually the opposite of hi-tech. He maintains a high degree of goofy aloofness from the technologies he writes about in such obsessive detail - almost as if just using them would increase the risk of being somehow "infected" by them. He wrote his most famous novel, Neuromancer, on a 1927 olive-green Hermes portable typewriter, and only recently migrated to a battered old Apple Mac." Gibson famously discussed his use of a typewriter in a phone interview for Playboy, August 30, 1996. "I do remember sitting with a blank sheet of paper and a typewriter going to 'dataspace' and 'infospace' and a couple of other clunkers, and then coming to 'cyberspace' thinking it sounds as though it means something." I have touched on the question of Gibson's and another influential contemporary novelist's apparently conservative approach to, shall we say, avant-garde practice in a relatively early online work, John Cayley, "Why Did People Make Things Like This?" [Web site] (Electronic Book Review, 1997 [cited February 2002]); available from ebr* Thus not only are the formal characteristics and the materiality of Gibson's language at odds with the flickering signification of its represented content, but, at the very least, the once-preferred experience of this writer - his phenomenology of inscription - is an apparent denial of Hayles' critical progression. I want to emphasise, in making these remarks, that if the subjective experience of the critic or reader is brought forward as evidence for a change in the structures of signification, then it is all the more important to examine the practices of the writer and the formal qualities of the work produced by those practices. Gibson sitting at a typewriter composing a novel may well produce a representation of the concept of flickering signification, but his practice does not necessarily embody the potential for new structures of meaning generation, or instantiate a corresponding materiality of language.

We will return to practice, but first I would like to examine Hayles' flickering signifier in so far as it engages with the notion of code-as-text. *Hayles, "Virtual Bodies and Flickering Signifiers," 31. The immediately following quotations, interspersed with my comments are from what I take to be a crucial paragraph in Hayles' crucial article.* "In informatics, the signifier can no longer be understood as a single marker, for example an ink mark on a page. Rather it exists as a flexible chain of markers bound together by the arbitrary relations specified by the relevant codes...." At least since Saussure, it seems somewhat redundant to point to the arbitrariness of any signifer-signified relation. I suppose that Hayles is actually referring to these relations as 'arbitrary' because they are not necessarily significant as human readings; they are not addressed to general human readers but only to the systems and systems-makers who have coded or specified them for certain purposes. They are, nonetheless, construable and are far from arbitrary when considered as addressed to the systems in which they are embedded. They have both significance and consequence. "...As I write these words on my computer, I see the lights on the video screen, but for the computer, the relevant signifiers are electronic polarities on disk...." That is, they are Kittler's (fundamental) signifiers of voltage difference. "...Intervening between what I see and what the computer reads are the machine code that correlates these symbols with binary digits, the compiler language that correlates these symbols with higher-level instructions determining how the symbols are to be manipulated, the processing program that mediates between these instructions and the commands I give the computer, and so forth. A signifier on one level becomes a signifier on the next-higher level." Hayles goes on to discuss the 'astonishing power' which these 'arbitrary,' hierarchically structured chains of codes generate, since manipulations, interpreted as commands at one level can have cascading, global effects. This is, presumably, 'power' in the now familiar technological sense, as used in the advertising and publicity for computer systems where, to relate the term with a more general or 'Foucauldian' sense, we may think of it as the power to alter the behaviour of a system in an impressive manner or at great speed, etc. By shifting the argument in this way, I think she has bracketed a more significant consequence of the structure of signification which she is delineating: the question of address, the address of the specific encoded 'levels.'

In an article on 'digital code and literary text,' Florian Cramer has pointed out that, as he somewhat obscurely puts it, "... the namespace of executable instruction code and nonexecutable code is flat." *Florian Cramer, Digital Code and Literary Text [Article in Web-based journal] (BeeHive Hypertext/Hypermedia Literary Journal, 2001 [cited February 2002]); available from http://beehive.temporalimage.com/content_apps43/app_d.html.* From the context his meaning is clear: that the same character or symbol set is used - for example - to transcribe both the text being word processed and (to be precise) the source code of the program which may be doing the word processing. On the level plains of letters and bits, there is no radical disjuncture in the symbolic media when we cross from a region of 'executable' text to text 'for human consumption.' From the human reader's point of view, they are both more or less construable strings of letters; from the processing hardware's point of view they are more or less construable sequences of voltage differences. On the one hand, this statement is related to the famous inter-media translatability of digitised cultural objects (once coded, regular procedures can be used to manipulate an image, a segment of audio, a text, etc. without distinction, disregarding the significance or affect of the manipulation). Cramer is, however, more concerned with the potential for sampling and mixing code and text (in the contemporary music sense). Again, as in Hayles' analysis, the question of the address of specific code segments and texts is bracketed. Not only is it bracketed, but the range of positions of address is simplified, as if we are speaking of a flat letterspace for: code on the one hand and text on the other; whereas, clearly, there are many levels. Both Cramer and Hayles recognize a multi-level hierarchy of codes without elaborating or distinguishing them in the course of their discussions. Within the field of networked and programmable media, at the very least, we can acknowledge: machine codes, tokenised codes, low-level languages, high-level languages, scripting languages, macro languages, markup languages, Operating Systems and their scripting language, the Human Computer Interface, the procedural descriptions of software manuals, and a very large number of texts addressed to entirely human concerns. *In passing it is worth highlighting the interface itself, particularly the ever-evolving HCI, as a complex programmable object with a structure like a language, including, in some cases, an underlying textual command-line interface which mirrors the now familiar mimetic and visual instantiation of users' interface. This is another point for potential artistic intervention as well as an vital consideration when discussing the emergent rhetorics of new media, as Manovich has demonstrated so well, even introducing the powerful concept of 'cultural interface' (human-computer-culture interface) as an analytic tool. Manovich, The Language of New Media 62-115.*

For Cramer, and not only for Cramer, this simplified, bracketed, or ambiguous textual address has become a valorised aesthetic and even a political principle: "...computers and digital poetry might teach us to pay more attention to codes and control structures coded into all language. In more general terms, program code contaminates in itself two concepts which are traditionally juxtaposed and unresolved in modern linguistics: the structure, as conceived of in formalism and structuralism, and the performative, as developed by speech act theory" (Cramer, Digital Code and Literary Text.) To attempt a paraphrase: working or sampled or intermixed or collaged code, where it is presented as verbal art, is seen by Cramer to represent, in itself, a revelation of underlying, perhaps even concealed, structures of control, and also (because of its origins in operative, efficacious program code) to instantiate a genuinely 'performative' textuality, a textuality which 'does' something, which alters the behaviour of a system. It has the 'astonishing power' of other cultural manifestations of new technology and new media, the power that Hayles has also recognized as a function of the coded structures arranged at various 'levels' in programmatological systems, chained together by a literal topography, which is 'flattened' by a shared symbol set. We should pause to consider what this power amounts to. What are the systems whose behaviour can be altered by this power?

In the criticism of theoretically sophisticated poetics there is a parallel aesthetic and political agenda, which I am tempted to call the Reveal Code Aesthetic. It is partly documented and particularly well-represented in, for example, Marjorie Perloff's Radical Artifice, where 'reveal code' is revealed as a project of L=A=N=G=U=A=G=E writers such as Charles Berstein, after having been properly and correctly situated in the traditions of process-based, generative and/or constrained literature and potential literature by Modernist, OuLiPian, Fluxus and related writers culminating, for Perloff, in John Cage and the L=A=N=G=U=A=G=E writers themselves. *Majorie Perloff, Radical Artifice: Writing Poetry in the Age of Media (Chicago: University of Chicago Press, 1991) 189. For a separate but related discussion of some of these issues, see John Cayley, Pressing the "Reveal Code" Key [Email-delivered, peer-reviewed periodical] (EJournal, 1996 [cited February 2002]); available from http://www.hanover.edu/philos/ejournal/archive/ej-6-1.txt. The work of Emmett Williams and Jackson Mac Low, central to any assessment of the radical poetic artifice which she identifies, as also for the criticism of writing in networked and programmable media, is notable for its absence from Perloff's book.* Although the political and aesthetic of program of 'reveal code' appears to be shared with Cramer's new media writers, in the context of Perloff's poetics, the codes revealed and deconstructed in language per se (rather than digitised textuality) are as much those of "the inaccessible system core," the machinic devices that conceal "the systems that control the formats that determine the genres of our everyday life." (Radical Artifice 188; Perloff is citing an earlier form of Charles Bernstein, "Play It Again, Pac-Man,") While the progressive tenor of an aesthetic and political deconstruction underlies this project, there is something of a Luddite tone in Perloff. *As more writers from this tradition make the move into 'new media,' this position begins to change. They become 'new media writers' 'digital poets,' etc. and attitudes perceptibly shift. Writers also, of course, become more sophisticated in their understanding of programmatological systems. This can be seen particularly in Charles Bernstein's subsequent writing on digital media and also, for example, in the work of Loss Pequeño Glazier, who is closely associated with the poetic practice which has developed from the L=A=N=G=U=A=G=E 'school.' See below, and, just-published, Loss Pequeño Glazier, Digital Poetics: The Making of E-Poetries (Tuscaloosa: University of Alabama Press, 2002). reviewed in ebr by Brandon Barr The critical history of this (anti-)tradition in poetic literature is generally traced at least back to Mallarmé. A convenient source for its study can be found in the two-volume anthology: Jerome Rothenberg and Pierre Joris, eds., Poems for the Millennium: The University of California Book of Modern and Postmodern Poetry, vol. 1: From Fin-de-Siècle to Negritude (Berkeley: University of California Press, 1995). Jerome Rothenberg and Pierre Joris, eds., Poems for the Millennium: The University of California Book of Modern and Postmodern Poetry,* *vol. 2: From Postwar to Millennium (Berkeley: University of California Press, 1998).* New media writers and artists necessarily have more ambiguous political and aesthetic relations with the control structures of the media which carry their work.

The code-revealing language artists discussed by Perloff, both in their work and in their performance - be it textual performance or performance art per se or activism or (academic) critical practice - represent far better examples of the instantiation of pattern/randomness (distinguished from presence/absence) than the novelists cited by Hayles, even including Burroughs or Pynchon. While retaining her focus on the contemporary or near-contemporary writers which she associates with an innovative, L=A=N=G=U=A=G=E-inflected poetics having avant-garde inclinations, Perloff recalls an extensive tradition of poetic literature which is marked both by its attention to the materiality of language and its radicalisation of poetic practices. Perloff invokes formations and works by individuals which are also referred to by critics of writing in networked and programmable media. Like Cramer, she discusses the OuLiPo (Ouvroir de Littérature Potentielle), the working group inspired and once led by Raymond Quenueau, which is, perhaps, the primary reference for literary projects which are explicitly concerned with the application of algorithmic procedures, arbitrary constraint, generative or potential literature, and (relatively early) experimentation with the use of software. In doing so, she directly confronts the 'repression' of 'numerical,' generative procedures in poetry and poetics and turns to the work of John Cage as a cross-media figurehead. While only a minor aspect of his oeuvre, as compared with his major contribution to the art of (musical) sound, Cage's mesostic texts, especially his 'reading through' of Pound and Joyce, stitch together a range of concerns - inter-media art, procedural composition, the rereading (and implicit deconstruction) of the High Modernists - which are highly relevant both to contemporary poetics and to writing in networked and programmable media. If Cage's work is recalled in the context of the Fluxus movement (with which he is associated), then its relevance widens and deepens. Fluxus is a model of performative art practice (including explicitly literary practice) where the record of inscription is problematized (the work is an event, or the publication of a set of materials which must be manipulated by the reader/user), and where the presence/absence dialectic has been side-stepped by representations which may literally absent an artist-author. Perloff does not discuss Fluxus at length and so misses the opportunity to reassess and contextualize work by two of the most important practitioners of the '(numerical) repressed,' Emmett Williams and Jackson Mac Low, both of whom deserve serious study as precursors if not 'anticipatory plagiarists' of writing in networked and programmable media. *The term is (ironic) OuLiPian, used of any prior instantiation of work generated by a procedure which has subsequently been invented and specified by the OuLiPo. Such a discussion is beyond the scope of this essay.* Fluxus also provides a historical, critical link to the traditions of visual and concrete poetics, which are discussed in Perloff's account, particularly relevant work by Steve McCaffery and Joanna Drucker. The materiality of this work, considered as language art, visibly demonstrates a radical engagement with linguistic media and a requirement for the reader to engage with the codes - textual, rhetorical, paratextual, visual, etc. - by and of which the work is constituted.

If such prior work remains inadequately acknowledged in the discussion and reassessment of 'codework,' this may be, in part, simply because the traces of its inscriptions are captured and recorded in the 'durable material substrates' of print culture. Whereas Lacan's 'floating signification' is read as an analytic metaphor, applied to language borne by a delivery media (print) on which the signs of the interface texts literally 'rest' (where they have been impressed) or, at best, 'interleave,' (they do not 'float'), we read Hayles' 'flickering signifiers' (as she encourages us to do) as literally 'flickering,' and constituent, as such, of text which has become 'screenic.' As such, it seems to exist elsewhere, not on the page but through the window of the screen in the informatic realm (Manovich, The Language of New Media 94-115. Undoubtedly, there are clear and historical distinctions of delivery media for text. Nonetheless, we must be careful to distinguish the effects of delivery media on signification and affect from those produced by shifts in the compositional media, and there is great congruence between the approach to compositional media of certain print-based writers (such as those discussed by Perloff, for example), and the potential use of compositional media which is suggested by new media, i.e. new delivery media. This potential of text- and language-making is not necessarily engaged simply because new delivery media happen to be employed. *The argument here is a rehearsal of the familiar but ever-important argument against art practice, particularly new media art practice, as media-specific or media-determined. Cramer's essay makes similar points*. The locus classicus for a multi-layered, multi-level code-inflected writing and reading is, of course, Barthes' S/Z, as Hayles explicitly acknowledges. *Roland Barthes, S/Z, trans. Richard Miller (Oxford: Blackwell Publishers, 1990). N. Katherine Hayles, How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics (Chicago: University of Chicago Press, 1999) 46.* S/Z was concerned with a short story programmed in 'a persistent material substrate' but Barthes was nonetheless able to demonstrate the potential for an iterative flickering of hermeneutic attention across structured linguistic codes, implying, I would argue, perfectly adequate complexity, mobility, and programmability in the compositional media. Barthes' essay, after all, was not a demand for new media but a (re)call to new or latent ways of reading and writing.

We turn, nonetheless, to examples of what Cramer calls 'codework.' Cramer cites (amongst others) some of those writers in networked and programmable media whose work I, too, would consider in this context: Mez, Talan Memmott, Alan Sondheim, Jodi (references to specific works are given below). Leaving Jodi to one side for the moment, these are all artists who both work with code and make coded, programmatological objects. They are particularly known and notable for working code and code elements into what we might call the 'interface text' (the words which are available to be read by the human audiences they address). *Although I do not make use of his analysis in this essay, it is well worth referring to Philippe Bootz's analyses of systems-mediated textuality, where I believe my 'interface text' roughly corresponds to his 'texte-à-voir.' See Philippe Bootz, "Le point de vue fonctionnel: point de vue tragique et programme pilote," alire 10 / DOC(K)S Series 3, no. 13/14/15/16 (1997).* The result is a language which seems to be - depending on your perspective - enlivened or contaminated by code. In the rhetoric of this type of artistic production, contamination or infection (see

Cramer as quoted above and Hayles below) is more likely to be the requisite association since transgression of the deconstructed systems of control is an implicit aspect of the aesthetic agenda. For the moment, however, we are more concerned with certain formal and material characteristics of the resulting language.

The language certainly reveals code and code elements, but what code does it reveal? What does it tell a code-naïve reader about the characteristics and the power of code? Is it, indeed, still code at all? At what level does it sit in the chained hierarchies of flickering signification? Has it been incorporated into the 'interface text' in a way which reflects its hierarchical origin, if it has one? Only if these and other questions can be given answers which specify how and why code is sampled in this writing would be it 'codework' in a strong sense. (Perhaps we should reserve Mez's 'code wurk' for the weaker sense of code-contaminated language.) In the case of all of these writers (we'll come to Jodi shortly) the code embedded in the interface text has ceased to be operative or even potentially operative. It is 'broken' in the now familiar programmer's jargon. The breakdown of its operations eliminates one aspect of its proposed aesthetic value and allure, its native performative efficacy (which Cramer identified as a final throwaway without actually demonstrating or elaborating): the power of code to change the behaviour of a system. The code-as-text is more in the way of decoration or rhetorical flourish, the baroque euphuism of new media. This is not to say that - as part of the interface text - it may not generate important significance and affect. In particular, the address of this type of intermixed, contaminated language is often concerned - as shown in the work of all of these writers - with issues of identity, gender, subjectivity, technology, technoscience, and the mutating and mutable influence they bring to bear on human lives and on human-human and human-machine relationships.

For the moment, however, we are more concerned with certain formal and material characteristics of the resulting language. In a recent conference paper, Hayles has discussed the language of Memmott's From lexia to perplexia in terms of pidgins and creoles. "In this work the human face and body are re-coded with tags in a pidgin that we might call, rather than hypertext markup language, human markup language. Code erupts through the surface of the screenic text, infecting English with programming languages and resulting in a creole discourse that bespeaks an origin always already permeated by digital technologies." *N. Katherine Hayles, "Bodies of Texts, Bodies of Subjects: Metaphoric Networks in New Media" (paper presented at the Digital Arts and Culture conference, Providence, RI, 2001). This is cited from the version of the paper posted in PDF form before the conference. Please note: Hayles may well have revised it since.* Similarly, Mez has characterized her textual production as written in a new "language/code system" which she calls 'mezangelle.' *Mez, _the Data][H!][Bleeding T.Ex][E][Ts [Website] (2001 [cited February 2002]); available from http://netwurkerz.de/mez/datableed/complete/index.htm. "the texts make use of the polysemic language/code system termed _mezangelle_, which evolved/s from multifarious email exchanges, computer code re:appropriation and net iconographs. to _mezangelle_ means to take words/wordstrings/sentences and alter them in such a way as to extend and enhance meaning beyond the predicted or the expected."* It is perhaps unfair to treat what may be metaphoric usages as literal; however, I believe this use of pidgin and creole is, in particular, a significant misdirection. A pidgin is a full-blown language, albeit arising from the encounter and hybridization of two or more existing languages; a creole is a pidgin which has become a first language for speakers raised by previous generations who have created or used a pidgin. The point here is that, in the case of a pidgin, the elements which combine to generate new language are commensurate - linguistic material is not simply being injected from one hierarchically and functionally distinct or programmatologically-operative symbolic sub-system (which is subsumed within a full-blown culture-bearing system of human language use) into another. The creation of a pidgin is, furthermore, the result of interactions by commensurate entities, i.e. humans. In the code-as-text which we have seen to date - in the texts of a reveal code aesthetic - human-specified code elements and segments are, typically, incorporated into what I have called the 'interface text' which is unambiguously and by definition an instance of some human-readable language. It may be contaminated, jargonized, disrupted language, but it is not a new language, not (yet) evidence for the invasion of an empire of machinic colonizers whose demands of trade and interaction require the creation of a pidgin by economically and linguistically disempowered human users. *Not '(yet)' as I say, although some might wish to try making the strong case for an emergent machinic culture, which is, I believe, a serious project although a misdirection in this context.* The codeworks currently available to us extend, infect, and enhance natural language, but they do not create, for example, Code Pidgin English. *As in the term 'Chinese Pidgin English.' Cf. for example, the discussion in Charles F. Hockett, A Course in Modern Linguistics (New York: Macmillan, 1958) 420-23.*

The code has ceased to function as code. The resulting text pretends an ambiguous address: at once to human reader and to machinic processor, but both human and machine must read the code as part of human discourse. We would not try to compile the code in the interface texts of Memmott, Mez or Sondheim. Nonetheless, this pretended ambiguity of address remains important to the aesthetics of this work. It assumes or encourages an investment on the part of its readers in the technology of new media and, especially, in the dissemination of textual art in new media. Thus, the experiences of the reader in these worlds can be brought to bear on their reading of the codework and they can appreciate, through more-or-less traditional hermeneutic procedures, the references and allusions to technology, technoscience and the issues with which they confront us. However, I would argue, if this pretended ambiguity of address exhausts the aesthetics and politics of a project (I am not saying that it does in any of these cases) then it leaves open questions of the work's affect and significance when compared, for example, with previous poetic work in more durable material and linguistic substrates, some of which has been cited above.

The work of Sondheim needs to be singled out, in terms of practice and form, since his use of code is well-integrated into a long-term and wide-ranging language art project. The print-media version of Jennifer, for example, reads more in the tradition of innovative or avant-garde writing than as subsumed within codework or a reveal code aesthetic. *Alan Sondheim, Jennifer (Salt Lake City: Nominative Press Collective, 1998). Eastern US: http://jefferson.village.virginia.edu/~spoons/internet_txt.html; western US: http://www.anu.edu.au/english/internet_txt; additional images: http://www.cs.unca.edu/~davidson/pix/. The internet references associated with this citation will lead on to Sondheim's wider project*. Most of the texts is this selection are manipulated language, but often using procedures which are not directly related to codes and processing. Thus, while his overt subject matter - mediated gender and sexuality, explicitly inflected by computing and technoscience - and his explicitly chosen media keep him immediately allied with codeworking colleagues, Sondheim's work must also be read against earlier and contemporary writers working within or with a sense of the formally and aesthetically innovative traditions of poetics, and not only the poetics which intersects with Burroughs and Acker. With the implication that Sondheim's writing needs to be judged as such and should not necessarily be granted a special credit of affect or significance because of its instantiation in new media.

In the necessity to read the work in both a programmatological context and in the broader context of innovative writing - though in this sense only - Sondheim's engagement rhymes momentarily with that of Loss Pequeño Glazier. Glazier and his work represents a literal and explicit embodiment of "a set of relationships - relationships constituted by artistic practice - between a newly problematized linguistic materiality and represented content." Glazier has produced a body of work, grounded in an existing writing practice, which has covered a wide range of potential forms for digital poetics and he has, moreover, documented and analysed this trajectory in a series of critical contributions. *Most recently in the book gathering many of these papers and essays. Glazier, Digital Poetics: The Making of E-Poetries. which, please note, includes a chapter devoted to "Coding Writing, Reading Code." Glazier's work has been done while he has also served as one of the motive forces and prime initiators of the major resource for innovative writing on the internet, the Electronic Poetry Center at the University of Buffalo, http://epc.buffalo.edu.* Glazier's work is characterized by his use of code and the language of code. In this, I believe, he affords himself significant ironies. He writes, for example: "The language you are breathing become the language you think... These are not mere metaphors but new procedures for writing. How could it be simpler? Why don't we all think in UNIX? If we do, these ideas are a file: I am chmoding this file for you to have read, write, and execute permission - and please grep what you need from this! What I am saying is that innovative poetry itself is best suited to grep how technology factors language and how this technology, writing, and production are as inseparable as Larry, Moe, and Curly Java" (Glazier, Digital Poetics 31-32). This is discursive prose of a kind, but it is infected or contaminated by both code and poetry. Glazier doesn't think in UNIX, nor would he ever wish to do so. But his language is not 'mere metaphor' (poetry is not metaphor) it is centred on language-making (what poetry is), and it demands a poetic practice which is alive to new procedures and new potential and which is sensitive to the changes this practice produces in the materiality of the language itself. Apart from its engagement with code and coding, Glazier's work is also characterized by its bilingualism, or rather the multi-lingualism of 'America' in the sense of a Latin America which exists as historical and political soul and shadow throughout, arguably, the greater part of the United States. I raise this point to highlight distinctions in the way we may choose to consider the non-standard-English material in Glazier's (and others') texts, (while recalling Hayles' metaphoric analysis via 'pidgin' and 'creole'). In a Glazier text, there is a use of English intensified by an address to the materiality of language. There is the incorporation (in a strong sense, sometime within the body of a word) of linguistic material from Spanish and other languages, especially those indigenous to Mexico. There is a similar incorporation of linguistic material from code and from computing jargons. *See: Loss Pequeño Glazier, "White-Faced Bromeliads on 20 Hectares" [Javascripted algorithmic text on website] (Electronic Poetry Center, 2001 [cited February 2002]); available from http://wings.buffalo.edu/epc/authors/glazier/java/costa1/00.html. This relatively recent work illustrates my specific points but also demonstrates that Glazier has been exploring the properly programmatological dimension of writing in networked and programmable media with, for example, kinetic and algorithmic texts. A classified selection of texts is at: http://epc.buffalo.edu/authors/glazier/e-poetry/.* But whereas the use of other natural language material evokes significance and affect which is commensurate with human concerns - personal, political, social and cultural history, etc. - the use of 'codewords' evokes other concerns, closer to questions of technology and the technology of language. Glazier would rather think in Nahuatl than in UNIX, but in practice he prefers to think in P=O=E=T=R=Y.

Jodi takes us to another point in the textonomy of code-as-text, a relatively extreme position where code-as-text is, perhaps, all there is. *Jodi is the very well-known, long-standing net.art project of Joan Heemskerk and Dirk Paesmans. Jodi, www.jodi.org [Website] (Jodi, 1980- [cited February 2002]); available from http://www.jodi.org.* It is difficult to say anything hard and fast in terms of more-or-less conventional criticism about a site which is hardly ever the same on successive visits. Instead, I want to refer to what I remember of a visit in which a dynamic html- and javascript-mediated experience proved to have been delivered by html source which was, itself, a work of ASCII art. *The practice of composing ASCII symbols, usually displayed as monospaced fonts for regularity, in order to generate imagery. In Jodi's case this was abstract or verging on the abstract whereas, popularly, ASCII art has been figurative.* Here, the actual code is a text, an artistic text. However, the code is not, in this instance, working code (at least not 'hard-working,' shall we say). It is comprised of code segments which are ignored in the browser's interpretation and rendering of the html. The syntax of this markup language is particularly easy to manipulate or appropriate in this way because comments (ignored by any interpreter, by definition) may be extensive and because interpreters (browsers in this case) are, typically, programmed to ignore any<tagged<thing which they cannot render. The code works, but it is not all working code. Again it represents only a pretended ambiguity of address: its primary structures of signification were never meant for a machine or a machinic process.

I, too, have made a few 'codeworks' of a not dissimilar kind. By extracting and manipulating segments of the close-to-natural-language, very-high-level, interpreted programming language, HyperTalk, I was able to make human-readable texts which are also segments of interpretable, working code:

```
on write
  repeat twice
    do "global " & characteristics
  end repeat
  repeat with programmers = one to always
    if touching then
      put essential into invariance
    else
      put the round of simplicity * engineering / synchronicity + one into invariance
    end if
    if invariance > the random of engineering and not categorical then
      put ideals + one into media
      if subversive then
        put false into subversive
      end if
      if media > instantiation then
        put one into media
      end if
    else
      put the inscription of conjunctions + one into media
    end if
    if categorical then put false into categorical
    put media into ideals
    put word media of field "text" of card understanding & "text" into potential
    if the mouse is down then
      put conjunctions into potential
      put potential into card field agents
      put true into encoded
      exit repeat
    end if
    inflect
    wait manipulation
    put potential into conjunctions
    put ideals into world
    if performed then put false into performed
    if programmers are greater than control and media & comma is in field computer of
card understanding & "text" then exit repeat
  end repeat
  if not encoded and not touching then
    if ideals are developed then wait five seconds
    lock screen
    put empty into card field agents
    put empty into card field system
    do "unlock screen with dissolve" & fantasies
  end if
end write
```

Cayley, Pressing the "Reveal Code" Key. The variable terms in this code were randomly and systematically replaced with substantive words from the text on which the procedure operates - any noun or adjective was allowed to replace a variable name containing a value; any verb replaced a procedure or function name. HyperTalk 'reserved words' were left intact. The code is working code.

This text has genuinely ambiguous address - to a HyperTalk interpreter and to human readers. It could (and does, in some versions of the software) alter the behaviour of a system, when included as one routine in a text generator. Its address to human concerns is clearly ludic and, perhaps, pretends more than it delivers in terms of significance and affect, but at least we can say, with little qualification, that this code is the text.

But where is such a codetext going, in terms, for instance, of its formal and rhetorical characteristics, in terms of its specific materiality? As a text - let us provisionally call it a poetic text in the sense of a text which implies some trial of language - which is addressed to human readers, it has distinct limitations, constraints which disallow or compromise its engagement with broader and more traditional concerns or sources of cultural value. Nonetheless, for me, it suggests new or newly highlighted rhetorical strategies which are specific to the materiality of language in networked and programmable media. For the moment, I will identify two such rhetorical fields of play: 1) the direct confrontation of strict logical-syntactic symbolic composition (programming per se) with natural language syntax and argument, and 2) what I think of as a potential 'aesthetics of compilation': the creation of linguistic or symbolic constructs which are designed, for example, to be read in one mode of address and at one level of code in a chained hierarchy of symbolic systems, while simultaneously intended for compilation into a systematically related code at a different level within the hierarchy, with a different mode of address.

The first of these rhetorical fields represents an age-old and persistent problem: that, indeed, of logic v. rhetoric, although recast in specific proliferating instances of logic-as-literature in new media. There is no time or space in this shortly closing essay to take this on. *But I raise it, in part, thanks to remarks by Nick Monfort to be published along with a forthcoming paper of mine, John Cayley, "Literal Art: Neither Lines nor Pixels but Letters," in First Person: New Media as Story, Performance, and Game, ed. Noah Wardrip-Fruin and Pat Harrigan (Cambridge: MIT Press, forthcoming 2002). As I was finishing this essay I came across Hugh Kenner's highly interesting reading of "Beckett Thinking." Kenner examines Beckett's writing in terms of strict, exhaustive logical procedure in an essay which includes paraphrases coded in the programming language Pascal. Hugh Kenner, The Mechanic Muse (Oxford: Oxford University Press,* *1987) 85-105*. Compilation in language and literature, however, directly addresses the interrelation of code and text; and it seems to me to be a good example of a rhetorical concept, hitherto of little use where literary objects were inscribed in persistent and durable material substrates, but of great potential in a literature constituted by flickering signification. Texts are already being made to be compiled, decompiled, recompiled, and so on.

I may have seemed to be arguing with flickering signification, by giving examples of writing which appeared to demonstrate its structures of code and text in systematically linked hierarchies, and then showing that these structures were collapsed in many of the examples to hand. In fact, I believe that the structures Hayles identifies and characterises are clearly operative in writing in networked and programmable media, just as they are operative in certain types of innovative poetic practice. The writing of flickering signification does, indeed, contribute to changes in the body of literature, the literary corpus, both its 'material substrate' and its 'codes of representation.' However, rather than the intermixing and mutual contamination of code and text, we require not only a maintenance and practical understanding of the distinction between code and text, we need at least the same range and fineness of distinction as that which exists among all the levels of programmatological languages and codes. The 'power' - including any affect and significance discoverable by interpretation - which such structures of signification generate is dependent on these distinctions, and on the compilation procedures (which I propose as rhetorical) by which they are systematically related. This 'power' is also, typically in this context, dependent on the concealment - the hidden working - of the code which is thus allowed to serve its function as program, to generate the text and offer it - iteratively, repeatedly, indeterminately - for instances of potential performance, including the familiar performance of reading.

In her discussion of the flickering signifier and its filial relation to the floating precursor of Lacan, mutation is the resultant process of a dialectic implied by a new structure of signification, a parallel term for castration in Lacan's analysis. Mutation is "a decisive event in the psycholinguistics of information. Mutation is the catastrophe in the pattern/randomness dialectic analogous to castration in the presence/absence dialectic" (Hayles, "Virtual Bodies and Flickering Signifiers," 33). Mutation - which evokes change, movement, the kinetic potential of text in new media, the mimetic engagement of literature with the culture of human time - mutation is indeed a generative catastrophe for 'literature' in the sense of immutable, authoritative corpus. As writing in networked and programmable media, language and literature mutate over time and as time-based art, according to programs of coded texts which are embedded and concealed in their structures of flickering signification. For the code to function as generator, as programmaton, as manipulator of the text, it must, typically, be a distinct part of the global textual system; it must be possible to recompile the codes as operative procedures, as aspects of live-art textual practice. The code is not the text.

*

And now, indulge your loop aesthetic and take the simple, ever-implicit link back to the paragraphs that introduced this critique. Read them as if they were some instance of a provisional conclusion.

works cited

Aarseth, Espen. Cybertext: Perspectives on Ergodic Literature. Baltimore and London: John Hopkins University Press, 1997.

Barthes, Roland. S/Z. Translated by Richard Miller. Oxford: Blackwell Publishers, 1990.

Bernstein, Charles. "Play It Again, Pac-Man." Postmodern Culture 2, no. 1 (1991): a reworking of 'Hot circuits: a video arcade,' American Museum of the Moving Image, 14 June-26 November 1989.

Bootz, Philippe. "Le point de vue fonctionnel: point de vue tragique et programme pilote." alire 10 / DOC(K)S Series 3, no. 13/14/15/16 (1997): 28-47.

Cayley, John. "Literal Art: Neither Lines nor Pixels but Letters." In First Person: New Media as Story, Performance, and Game, edited by Noah Wardrip-Fruin and Pat Harrigan. Cambridge: MIT Press, forthcoming 2002.

------. "Pressing the 'Reveal Code' Key" [Email-delivered, peer-reviewed periodical]. EJournal, 1996 [cited February 2002]. Available from http://www.hanover.edu/philos/ejournal/archive/ej-6-1.txt.

------. "Why Did People Make Things Like This?" [Web site]. Electronic Book Review, 1997 [cited February 2002].

Cramer, Florian. "Digital Code and Literary Text" [Article in Web-based journal]. BeeHive Hypertext/Hypermedia Literary Journal, 2001 [cited February 2002]. Available from http://beehive.temporalimage.com/content_apps43/app_d.html.

Cunningham, Michael. The Virtual Tourist [a Short Interview with William Gibson] [Web site]. P45.net, 1996 [cited February 2002]. Available from http://www.p45.net/dos_prompt/columns/3.html.

Glazier, Loss Pequeño. Digital Poetics: The Making of E-Poetries. Tuscaloosa: University of Alabama Press, 2002.

------. "White-Faced Bromeliads on 20 Hectares" [Javascripted algorithmic text on website]. Electronic Poetry Center, 2001 [cited February 2002]. Available from http://wings.buffalo.edu/epc/authors/glazier/java/costa1/00.html.

Hayles, N. Katherine. "Bodies of Texts, Bodies of Subjects: Metaphoric Networks in New Media." Paper presented at the Digital Arts and Culture conference, Providence, RI 2001.

------. How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics. Chicago: University of Chicago Press, 1999.

------. "Virtual Bodies and Flickering Signifiers." In Electronic Culture: Technology and Visual Representation, edited by Timothy Druckery, 259-77. New York: Aperture, 1996.

------. "Virtual Bodies and Flickering Signifiers." In How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics, 25-49. Chicago: University of Chicago Press, 1999.

Hockett, Charles F. A Course in Modern Linguistics. New York: Macmillan, 1958.

Jodi. www.jodi.org [Website]. Jodi, 1980- [cited February 2002]. Available from http://www.jodi.org.

Kenner, Hugh. The Mechanic Muse. Oxford: Oxford University Press, 1987.

Kittler, Friedrich. "There Is No Software." In Literature Media Information Systems, edited by John Johnston, 147-55. Amsteldijk: G+B Arts International, 1997.

Manovich, Lev. The Language of New Media. Edited by Roger F. Malina, Leonardo. Cambridge: MIT Press, 2001.

Mez. _the Data][H!][Bleeding T.Ex][E][Ts [Website]. 2001 [cited February 2002]. Available from http://netwurkerz.de/mez/datableed/complete/index.htm.

Perloff, Majorie. Radical Artifice: Writing Poetry in the Age of Media. Chicago: University of Chicago Press, 1991.

Rothenberg, Jerome, and Pierre Joris, eds. Poems for the Millennium: The University of California Book of Modern and Postmodern Poetry. Vol. 2: From Postwar to Millennium. Berkeley: University of California Press, 1998.

------, eds. Poems for the Millennium: The University of California Book of Modern and Postmodern Poetry. Vol. 1: From Fin-de-Siècle to Negritude. Berkeley: University of California Press, 1995.

Sondheim, Alan. Jennifer. Salt Lake City: Nominative Press Collective, 1998.

Ulmer, Gregory L. Applied Grammatology: Post(E)-Pedagogy from Jacques Derrida to Joseph Beuys. Baltimore: John Hopkins University Press, 1985.

William Gibson (1948-) [Web site]. Guardian Unlimited, 2001 [cited February 2002]. Available from http://books.guardian.co.uk/authors/author/0,5917,96528,00.html.

Cite this Essay:

Cayley, John. “The Code is not the Text (Unless It Is the Text)” , Electronic Book Review, September 10, 2002, .

Readers wishing to respond to an essay in *ebr* may send ripostes or short glosses to the journal’s Managing Editor, Will Luers.

This essay was peer-reviewed.

At a Heightened Level of Intensity: A Discussion of the Philosophy and Politics of Language in John Cayley’s Digital Poetics

Auratures at the End(s) of Electronic Literature

Bass Resonance

Beginning with “The Image” in*How It Is*when translating certain processes of digital language art

Grammalepsy: An Introduction

electropoetics:

electropoetics

The Haunting of Benjamin Britten

Being Inside the Sentence

Poetry and Stuff: A Review of*#!*

by John Cayley

Not Just a River

by Rob Swigart