

What is code?

A conversation with Deleuze, Guattari and code

David M. Berry

Jo Pawlik

Code

The two of us wrote this article together. Since each of us was several, there was already quite a crowd. We have made use of everything that came within range, what was closest as well as farthest away. We have been aided, inspired multiplied [1].

JP: Code is described as many things: it is a cultural logic, a machinic operation or a process that is unfolding. It is becoming, today's hegemonic metaphor; inspiring quasi-semiotic investigations within cultural and artistic practice (e.g. The Matrix). *No-one leaves before it has set its mark on them...*

DB: Yes, it has become a narrative, a genre, a structural feature of contemporary society, an architecture for our technologically controlled societies (e.g. Lessig) and a tool of technocracy and of capitalism and law (Elul/Winner/Feenberg). It is both metaphor and reality, it serves as a translation between different discourses and spheres, DNA code, computer code, code as law, cultural code, aristocratic code, encrypted code (Latour).

JP: *Like the code to nourish you? Have to feed it something too.*

DB: Perhaps. I agree that code appears to be a defining discourse of our postmodernity. It offers both explanation and saviour, for example, the state as machine, that runs a faulty form of code that can be rewritten and re-executed. The constitution as a microcode, law as code. Humanity as objects at the mercy of an inhuman code.

JP: True and it gathers together a disturbing discourse of the elect. Code as intellectual heights, an aristocratic elect who can free information and have a wisdom to transform society without the politics, without nations and without politicians. Code becomes the lived and the desired. Both a black box *and* a glass box. Hard and unyielding and simultaneously soft and malleable.

Code as walls and doors of the prisons
and workhouses of the 21st Century

DB: Code seems to follow information into a displaced subjectivity, perhaps a new and startling subject of history that is merely a reflection of the biases, norms and values of the coding elite. More concerning, perhaps, code as walls and doors of the prisons and workhouses of the 21st Century. Condemned to make the *amende honorable* before the church of capital.

JP: So, we ask what is code? Not expecting to find answers, but rather to raise questions. To survey and map realms that are yet to come (AO:5). The key for us lies in code's connectivity, it is a semiotic-chain, rhizomatic (rather like a non-hierarchical network of nodes) and hence our map must allow for it to be interconnected from anything to anything. In this investigation, which we know might sometimes be hard to follow, our method imitates that outlined by Deleuze & Guattari in *Anti-Oedipus* (2004). It will analyse by decentering it onto other dimensions, and other registers (AO:8).

We hope that you will view this article as a “little machine” (AO: 4), itself something to be read slowly, or fast, so that you can take from it whatever comes your way. It does not ask the question of where code stops and the society starts, rather it forms a tracing of the code-society or the society-code.

DB: Dystopian and utopian, both can cling like Pincher Martin to code. Code has its own apocalyptic fictions; crashes and bugs, Y2K and corruption. It is a fiction that is becoming a literary fiction (Kermode). We wish to stop it becoming a myth, by questioning code and asking it uncomfortable questions. But by our questioning we do not wish to be considered experts or legislators, rather we want to ask again who are the “Gods” of the information age (Heidegger). By drawing code out and stretching it out, we hope to make code less mysterious, less an “unconcealment that is concealed” (Heidegger).

JP: Perhaps to ask code and coders to think again about the way in which they see the world, to move from objects to things, and practice code as poetry (poesis). Rather than code as ordering the world, fixing and overcoding. Code as a craft, “bringing-forth” through a showing or revealing that is not about turning the world into resources to be assembled, and reassembled forever.

DB: And let us not forget the debt that code owes to war and government. It has a bloody history, formed from the special projects of the cold war, a technological race, that got mixed up with the counter-culture but still fights battles on our behalf. *He laid aside his sabre. And with a smile he took my hand.*

Code as concept

DB: A stab in the dark. To start neither at the beginning or the end, but in the middle: code is pure concept instantiated into the languages of machines. Coding is the art of forming, inventing and fabricating structures based on these languages. Structures that constrain use as well as free. The coder is the friend of the code, the potentiality of the code, not merely forming, inventing and fabricating code but also desiring. *The electric hymn book that Happolati invented. With electric letters that shine in the dark?*

JP: And what of those non-coders who use code, or rather are used by code instead of forming it? Code can enable but it can also repress. Deleuze believes that we live in a society

Deleuze



of control and that code is part “of the numerical language of control” requiring of us passwords, user names, and the completion of form fields to either grant or deny access to information, goods and services (1992).

DB: Yes, code becomes the unavoidable boundary around which no detour exists in order to participate fully in modern life. It is ubiquitous. Formatted by code, harmonised with the language of machines, our life history, tastes, preferences and personal details become profiles, mailing lists, data and ultimately markets. Societies of control regulate their population by ensuring their knowing and unknowing participation in the marketplace through enforced compatibility with code. *Watch over this code!... Let me see some code!*

JP: But there is no simple code. Code is production and as such is a machine. Every piece of code has components and is defined by them. It is a multiplicity although not every multiplicity is code. No code is a single component because even the first piece of code draws on others. Neither is there code possessing all components as this would be chaos. Every piece of code has a regular contour defined by the sum of its components. The code is whole because it totalises the components, but it remains a fragmentary whole.

DB: Code aborescent. Plato’s building agile, object-oriented and postmodern codes under the spreading chestnut tree.

JP: But computers are not the only machines that use code. Deleuze believes that everything is a machine, or to be more precise every machine is a machine of a machine. By this

he means that every machine is connected to another by a flow—whether this flow is air, information, water, desire etc—which it interrupts, uses, converts and then connects with another machine.

DB: I agree that human beings are nothing more than an assemblage of several machines linked to other machines, though century's worth of history have us duped into thinking otherwise.

JP: But, does every machine have a code built into it which determines the nature of its relations with other machines and their outputs? How else would we know whether to swallow air, suffocate on food or drink sound waves? There is even a social machine, who's task it is to code the flows that circulate within it. To apportion wealth, to organise production and to record the particular constellation of linked up flows that define its mode of being.

DB: Up to this point, code is verging towards the deterministic or the programmatic, dependent upon some form of Ur-coder who might be synonymous with God, with the Despot, with Nature, depending on to whom you attribute the first and last words.

JP: But Deleuze delimits a way of scrambling the codes, of flouting the key, which enables a different kind of de/encoding to take place and frees us from a pre-determined input-output, $a=b$ matrix. Enter Desire. Enter Creativity. Enter the Schizo. Enter capitalism? *You show them you have something that is really profitable, and then there will be no limits to the recognition of your ability.*

Code as Schizo

DB: Deleuze & Guattari warned us that the Schizo ethic was not a revolutionary one, but a way of surviving under capitalism by producing fresh desires within the structural limits of capitalism. Where will the revolution come from?

JP: It will be a decoded flow, a “deterritorialised flow that runs too far and cuts too sharply”. D & G hold that art and science have a revolutionary potential. Code, like art and science, causes increasingly decoded and deterritorialised flows to circulate in the socius. To become more complicated, more saturated. *A few steps away a policeman is observing me; he stands in the middle of the street and doesn't pay attention to anything else.*

DB: But, code is bifurcated between a conceptual and a functional schema, an “all encompassing wisdom [=code]”.

Guattari



Concepts and functions appear as two types of multiplicities or varieties whose natures are different. Using the Deleuzian concept of Demon which indicates, in philosophy as well as science, not something that exceeds our possibilities but a common kind of these necessary intercessors as respective “subjects” of enunciation: the philosophical friend, the rival, the idiot, the overman are no less demons that Maxwell's demon or than Einstein's or Heseinberg's observers. (WIP: 129). *Our eyes meet as I lift my head; maybe he had been standing there for quite a while just watching me.*

JP: Do you know what time it is?

HE: Time? Simple Time?... Great time, mad time, quite bedivelled time, in which the fun waxes fast and furious, with heaven-high leaping and springing—and again, of course, a bit miserable, very miserable indeed, I not only admit that, I even emphasise it, with pride, for it is sitting and fit, such is artist-way and artist-nature.

Code and sense perception

DB: In code the role of the partial coder is to perceive and to experience, although these perceptions and affections might not be those of the coder, in the currently accepted sense, but belong to the code. Does code interpolate the coder, or only the user? Ideal partial observers are the perceptions or sensory affections of code itself manifested in functions and “functives”, the code crystallised affect.

JP: Maybe the function in code determines a state of affairs, thing or body that actualises the virtual on a plane of reference and in a system of co-ordinates, a dimensional classification; the concept in code expresses an event that gives

consistency to the virtual on a plane of immanence and in an ordered form.

DB: Well, in each case the respective fields of coding find themselves marked out by very different entities but that nonetheless exhibit a certain analogy in their task: a problem. Is this a world-directed perspective—code as an action facing the world?

JP: Does that not consist in failing to answer a question? In adapting, in co-adapting, with a higher taste as problematic faculty, are corresponding elements in the process being determined? Do we not replicate the chains of equivalence, allowing the code, to code, so to speak, how we might understand it?

DB: Coders are writers, and every writer is a sellout. *But an honest joy/Does itself destroy/For a harlot coy.*

JP: We might ask ourselves the following question: is the software coder a scientist? A philosopher? Or an artist? Or a schizophrenic?

Like the “Matrix”. Here is a film which has latched onto the concept of code and also its discussion in contemporary philosophy, almost smugly displaying its dexterity in handling both

AL: For me the only code is that which places an explosive device in its package, fabricating a counterfeit currency. *Which in part the knowing children sang to me.*

Dr. K: *This man is mad. There has been for a long time no doubt of it, and it is most regrettable that in our circle the profession of alienist is not represented. I, as a numismatist, feel myself entirely incompetent in this situation.*

DB: For Deleuze, the ascription of these titles exceeds determining whether the tools of the trade in question are microscopes and test-tubes, cafés and cigarettes, or easels and oil-paints. Rather they identify the kind of thinking that each group practices. Latour claimed that if you gave him a laboratory he could move the world. Maybe prosopopoeia is part of the answer, he should ask code what it thinks.

JP: But not just the kind of thinking, but the kind of problems which this thought presupposes, and the nature of the solutions that it can provide. To ask under which category the coder clicks her mouse is to question whether she is cre-

ating concepts as opposed to dealing in functions like a scientist, or generating percepts and affects like an artist.

DB: If you’re actually going to love technology, you have to give up sentimental slop, novels sprinkled with rose water. All these stories of efficient, profitable, optimal, functional technologies.

JP: Who said I wanted to love technology?

DB: The philosopher loves the concept. The artist, the affect. Do the coders love the code?

JP: If we say that code is a concept, summoning into being or releasing free software as an event, the coder is cast first and foremost as a philosopher. The coder, as philosopher, could neither love nor covet her code prior to its arrival. It must take her by surprise. For the philosopher, or more specifically the conceptual personae through whom concepts come to pass and are given voice, (Deleuze does not strictly believe in the creativity of an individual ego), Deleuze reserves a privileged role in the modern world which is so woefully lacking in creation and in resistance to the present. He writes: “The creation of concepts in itself calls for a future form, for a new earth and people that do not yet exist” (1994, 108). Deleuze would hope this future form would be recognizable by virtue of its dislocation from the present.

DB: If the software coder really is a philosopher, what kind of a future is free software summoning and who are the new people who might later exist?

But, does every machine have a code built into it which determines the nature of its relations with other machines and their outputs?

JP: Thanks to computers, we now know that there are only differences of degree between matter and texts. In fact, ever since a literary happy few started talking about “textual machines” in connection with novels, it has been perfectly natural for machines to become texts written by novelists who are as brilliant as they are anonymous (Latour). But then is there no longer any difference between humans and nonhumans.

DB: No, but there is no difference between the spirit of machines and their matter, either; they are souls through and through (Latour).

Code



JP: But don't the stories tell us that machines are purported to be pure, separated from the messy world of the real? Their internal world floating in a platonic sphere, eternal and perfect. Is the basis of their functioning deep within the casing numbers ticking over numbers, overflowing logic registers and memory addresses?

DB: I agree. Logic is often considered the base of code. Logic is reductionist not accidentally but essentially and necessarily; it wants to turn concepts into functions. In becoming propositional, the conceptual idea of code loses all the characteristics it possessed as a concept: its endoconsistency and its exoconsistency. This is because of a regime of independence that has replaced that of inseparability, the code has enframed the concept.

Code as science

DB: Do you think a real hatred inspires logic's rivalry with, or its will to supplant, the concept? Deleuze thought "it kills the concept twice over".

JP: The concept is reborn not because it is a scientific function and not because it is a logical proposition: it does not belong to a discursive system and it does not have a reference. The concept shows itself and does nothing but show itself. Concepts are really monsters that are reborn from their fragments.

DB: But how does this relate to the code, and more specifically to free software and free culture? Can we say that this is that summoning? Can the code save us?

JP: Free software knows only relations of movement and rest, of speed and slowness, between unformed, or relatively unformed, elements, molecules or particles borne away by

fluxes. It knows nothing of subjects but rather singularities called events or hecceities. Free software is a machine but a machine that has no beginning and no end. It is always in the middle, between things. Free software is where things pick up speed, a transversal movement, that undermines its banks and accelerates in the middle. But that is not to say that capital does not attempt to recode it, reterritorialising its flows within the circuits of capital.

DB: A project or a person is here only definable by movements and rests, speeds and slowness (longitude) and by affects, intensities (latitude). There are no more forms, but cinematic relations between unformed elements; there are no more subjects but dynamic individuations without subjects, which constitute collective assemblages. Nothing develops, but things arrive late or in advance, and enter into some assemblage according to their compositions of speed. Nothing becomes subjective but haecceities take shape according to the compositions of non-subjective powers and effects. Maps of speeds and intensities (e.g. Sourceforge).

JP: We have all already encountered this business of speeds and slowness: their common quality is to grow from the middle, to be always in-between; they have a common imperceptible, like the vast slowness of massive Japanese wrestlers, and all of a sudden, a decisive gesture so swift that we didn't see it.

DB: Good code, Bad code. Deleuze asks: "For what do private property, wealth, commodities, and classes signify?" and answers: "The breakdown of codes" (AO, 218). Capitalism is a generalized decoding of flows. It has decoded the worker in favour of abstract labour, it has decoded the family, as a means of consumption, in favour of interchangeable, faceless consumers and has decoded wealth in favour of abstract, speculative, merchant capital. In the face of this, it is difficult to know if we have too much code or too little and what the criteria might be by which we could make qualitative distinctions between one type of code and another, such as code as concept and code as commodity.

JP: We could suggest that the schizophrenic code (i.e. the schizophrenic coding as a radical politics of desire) could seek to de-normalise and de-individualise through a multiplicity of new, radical collective arrangements against power. Perhaps a radical hermeneutics of code, code as locality and place, a dwelling.

DB: Not all code is a dwelling. Bank systems, facial recognition packages, military defence equipment and govern-

mental monitoring software is code but not a dwelling. Even so, this code is in the domain of dwelling. That domain extends over this code and yet is not limited to the dwelling place. The bank clerk is at home on the bank network but does not have shelter there; the working woman is at home on the code but does not have a dwelling place there; the chief engineer is at home in the programming environment but does not dwell there. This code enframes her. She inhabits them and yet does not dwell in them.

Code as art

JP: You are right to distinguish between code as “challenging-forth” (Heidegger) and code that is a “bringing-forth”. The code that is reterritorialised is code that is proprietary and instrumental, has itself become a form of “standing-reserve”.

DB: So how are we to know when code is a “bringing-forth”? How will we know if it is a tool for conviviality. How will we distinguish between the paranoid and the schizophrenic?

JP: We know, that the friend or lover of code, as claimant does not lack rivals. If each citizen lays claim to something then we need to judge the validity of claims. The coder lays claim to the code, and the corporation, and the lawyer, who all say, “I am the friend of code”. First it was the computer scientists who exclaimed “This is our concern, we are the scientists!”. Then it was the turn of the lawyers, the journalists and the state chanting “Code must be domesticated and nationalised!” Finally the most shameful moment came when companies seized control of the code themselves “We are the friends of code, we put it in our computers, and we sell it to anyone”. The only code is functional and the only concepts are products to be sold. But even now we see the lawyers agreeing with the corporations, we must control the code, we must regulate the code, the code must be paranoid.

DB: This is perhaps the vision offered by William Gibson’s *Neuromancer*, a dystopian realization of the unchecked power of multinational corporations which, despite the efforts of outlaw subcultures, monopolize code. Through their creation of AI entities code becomes autonomous, it exceeds human control. If indeed it makes sense to retain the term human, which Gibson pejoratively substitutes with “meat”. The new human-machinic interfaces engendered by soft-

ware and technological development demand the jettisoning of received categories of existence as they invent uncanny new ones.

JP: This is the possibility of code. The code as a war machine. Nomadic thought. The code as outsider art, the gay science, code as desiring-production, making connections, to ever new connections.

 We might ask ourselves the following
 question: is the software coder a
 scientist? A philosopher? Or an artist?
 Or a schizophrenic?

DB: Code can be formed into networks of singularities into machines of struggle. As Capital de-territorializes code there is the potential through machines to re-territorialize. Through transformative constitutive action and network sociality—in other words the multitude—code can be deterritorializing, it is multiplicity and becoming, it is an event. Code is becoming nomadic.

JP: This nomadic code upsets and exceeds the criteria of representational transparency. According to Jean Baudrillard, the omnipresence of code in the West—DNA, binary, digital—enables the production of copies for which there are no originals. Unsecured and cut adrift from the “reality” which representation has for centuries prided itself on mirroring, we are now in the age of simulation. The depiction of code presents several difficulties for writers, who, in seeking to negotiate the new technological landscape, must somehow bend the representational medium of language and the linear process of reading to accommodate the proliferating ontological and spatio-temporal relations that code affords.

DB: This tension is as palpable in Gibson’s efforts to render cyberspace in prose (he first coined the term in *Neuromancer*) as it is on the book cover, where the flat 2D picture struggles to convey the multi-dimensional possibilities of the matrix. The aesthetics of simulation, the poetics of cyberspace and of hyperreality are, we might say, still under construction.

JP: Perhaps code precludes artistic production as we know it. Until the artist creates code and dispenses with representational media altogether, is it possible that her work will

contribute only impoverished, obsolete versions of the age of simulation?

DB: Artists have responded to “code” as both form and content. As form, we might also think of code as “genre”, the parodying of which has become a staple in the post-modern canon. Films such as “The Scream” series, “The Simpsons”, or “Austin Powers”; flaunt and then subvert the generic codes upon which the production and interpretation of meaning depends. More drastically, Paul Auster sets his “New York Trilogy” in an epistemological dystopia in which the world does not yield to rational comprehension as the genre of detective fiction traditionally demands. If clues are totally indistinguishable from (co)incidental detail, how can the detective guarantee a resolution, how can order be restored? As Auster emphasizes, generic codes and aesthetic form underwrite ideological assumptions and can be described as the products of specific social relations.

We know, that the friend or lover of code,
as claimant does not lack rivals

JP: And what of code as content? Like the “Matrix”. Here is a film which has latched onto the concept of code and also its discussion in contemporary philosophy, almost smugly displaying its dexterity in handling both.

DB: Or “I ♥ Huckabees” with its unfolding of a kind of existential code that underlies human reality. Are our interpretations shifting to an almost instrumental understanding of code as a form of weak structuralism? Philosophy as mere code, to be written, edited and improved, turned into myth so that our societies can run smoothly.

JP: The hacker stands starkly here. If code can be hacked, then perhaps we should drop a monkey-wrench in the machine, or sugar in the petrol tank of code? Can the philosopher be a model for the hacker or the hacker for the philosopher? Or perhaps the hacker, with the concentrations on the smooth, efficient hacks, might not be the best model. Perhaps the cracker is a better model for the philosophy of the future. Submerged, unpredictable and radically decentred. Outlaw and outlawed.

DB: Perhaps. But then perhaps we must also be careful of the fictions that we both read and write. And keep the radical potentialities of code and philosophy free.

*Wet with fever and fatigue we can now look toward the shore
and say goodbye to where the windows shone so brightly.*

Notes

[1] *We were, in fact, at least four, and we think you can guess who the others were.*

We hope to make code less mysterious

Bibliography

Deleuze, G. (1990). *Postscript on the Societies of Control*. L'autre Journal, Nr. 1.

Deleuze, G. (2004). *Foucault*. London: Continuum.

Deleuze, G., & Guattari, F. (1994). *What is Philosophy?* London: Verso.

Deleuze, G., & Guattari, F. (2004). *Anti-Oedipus: Capitalism and Schizophrenia*. London: Continuum.

Deleuze, G., & Guattari, F. (2003). *A Thousand Plateaus: Capitalism and Schizophrenia*. London: Continuum.

Copyright information

© 2005 David M. Berry, Jo Pawlik

This article is made available under the “Attribution-Share-alike” Creative Commons License 2.0 available from <http://creativecommons.org/licenses/by-sa/2.0/>.

About the author

David Berry is a researcher at the University of Sussex, UK and a member of the research collective The Libre Society (<http://www.libresociety.org>). He writes on issues surrounding intellectual property, immaterial labour, politics, free software and copyleft.

Jo Pawlik is a doctoral student at the University of Sussex researching the interaction between the American counterculture and French poststructuralism, focusing in particular on the deployment and political purchase of the concepts of madness and schizophrenia.