

# User:Andre Castro/WritingResearch/essay01

## From NetworkedMedia

< User:Andre Castro

### Contents

- 1 The Performativity of Code
  - 1.1 Outline
    - 1.1.1 Code as Language
    - 1.1.2 Code performativity
    - 1.1.3 Consent to perform
    - 1.1.4 Code and the individual
    - 1.1.5 Conclusion
  - 1.2 Abstract
  - 1.3 Code as Language
  - 1.4 Code performativity
  - 1.5 Consent to perform
  - 1.6 Code and the individual
  - 1.7 Conclusion
  - 1.8 Bibliography

## The Performativity of Code

Revised Version

### Outline

#### Code as Language

- code as a subset of natural languages

#### Code performativity

- speech-acts
- code - an unmediated performance

#### Consent to perform

- examples of our trust in code
- digitization:
  - reality becomes what fits into the digital representation of reality
  - work value - measured/accountable into 1 and 0
- digital = portable = spreadable; Consequences: its range become global

#### Code and the individual

- code shaping our subjectivity
- if not digital stored is not real - human senses and memory devalued
- interpersonal relations changed by social media
- comprehension of the effects of code - hard to assess, since we are immersed in them

## Conclusion

- reiterate: data are thus humanized and subjectivity is computerized
  - necessity to program
  - educational system disregarding code
  - agree: necessity to program
- 

## Abstract

In this essay I would like to investigate the performativity of code, what it means to say that code is performative. In order to do so I will begin by looking into the cross-relations between code and 'natural' languages. Once this relation is clear, I will begin the discussion by referring to performative speech-acts and attempt to gain an understanding of the differences between these and code performativity. Further on I will seek to define the conditions which allow code to become performative and give a glimpse of how our world and ourselves are both being transformed by code. The writings of Katherine Hayles, Florian Cramer and Jon McKenzie will constitute the main references for this short investigation. To conclude the essay, I will consider Douglas Roushokoff's vision of a future, where code becomes a power instrument that we must understand and employ, or else it will be performed upon us, if it does not already do so.

## Code as Language

Can the code from which software is built be considered a language? Florian Cramer in his essay *Language* states that "computer control languages are a formal (and as such rather primitive) subset of common human languages" (Cramer, 2008, p.168). Cramer bases his position on a series of shared characteristics of both languages. One of the most relevant points in this argument is the fact that common human languages, due to being culturally constructed by humans, are as artificial as computer programming languages. Furthermore, high-level computer programming languages stand in an intermediate position, between computers and humans, having to be comprehensible to both. For these reasons computer programming languages can never be too far from written language, otherwise they would just become incomprehensible to humans. \*

\* Note: In order to make the terminology less prone to confusions, from here onwards I will refer to computer control languages as code and common languages as English as in written and spoken language.

## Code performativity

According to Katherine Hayles, in her book *My Mother Was a Computer*, code exceeds both written and spoken language. "[C]ode that runs on a machine is performative in a much stronger sense than that attributed to language" (Hayles, 2005, p.50). By performative Hayles refers to the capacity that language possesses to act upon the world, producing what in linguistics is referred as performative speech-acts. Common examples of speech-acts are a judge convicting a person guilty of a crime, or a priest pronouncing a couple husband and wife. Although both cases illustrate situations in which speech-acts result in a radical change of their actors' lives, that change is actualized only through a system of agreements, which we obey to, otherwise speech-acts would be rendered irrelevant, simply words. Computer code, on the other hand, apart from altering the behavior of the machine in which it runs, has a direct impact upon the world. As Alexander Galloway puts it: "The imperative voice ... attempts to affect through persuasion but has little real material effect. So code is the first language that actually does what it says" (Galloway, 2004, pp: 165-166). Florian Cramer does not go that far in his perspective of code. For Cramer computer code does not have real and material effects on its own, being its range of action circumscribed to the machine. Nevertheless, when placed in a prominent position, where its behavior is considered as being always correct and is rarely given a close analytical look, as in the following section's examples, code is given the means to act upon the world:

"Computer languages become performative only through the social impact of the processes they trigger, specially when their outputs aren't critically checked ... as in the 1987 New York Stock Exchange crash that involved a chain traction of 'sell' recommendations by day trading software" (2008 pp. 170-171).

Code simply bypasses authoritative mediation, it does not need to be validated by a church, a court or a government, it is simply given green-light to perform upon the world in a very direct manner.

## Consent to perform

Why do we allow code to act upon reality without questioning it? Why was the software output that led to the 1987 Stock Exchange crash not considered susceptible to errors in very complicated scenarios? Why do we think of the GPS as a flawless technology and allow it to navigate us blindly, often leading to surprises when the final destination is reached? Or why do most of us do not question Google search mechanism in determining what information arrives to us at each search?

A possible understanding for the reasons which allow code to occupy such a prominent position in our contemporary world might emerge by looking at Florian Cramer's description of the digitization process and Jon McKenzie's perspective on knowledge in the postmodern world. Cramer refers to the computer as "a symbolic machine that computes syntactical language and processes alphanumeric symbols; it treats all data - including images and sounds - as textual, that is, chunks of coded symbols" (Cramer, 2008, p.171). In other words, reality, in order to be stored on a computer's hard-disk needs to be encoded into discreet syntactical units constituted by 0s and 1s, information that surpasses the encoding limitations is considered noise and therefore dumped. Jon McKenzie sees a similar encoding process taking place in reference to knowledge: In postmodernity knowledge has become measurable in terms of operational efficiency, which "demands that all knowledge must be translatable by and accountable in the '1's and '0's of digital matrices" (McKenzi, 2001, p.14). Being this the context in which we are in it is not surprising that we have given code permission to perform upon our world. Code has become the referee that determines what information and knowledge is, or in other words what is relevant to encoded and what is accessory and disposable. If code constitutes such a powerful authority, why would we contest its outputs, they will most certainly be correct and devised for our own good.

McKenzie also argues that "the digitization of discourses and practices enables them to be recorded, edited and played back in new and uncanny ways" (p.21). And by discourse I understand not only an aural or visual discourse, but also a written discourse, such as code, which can easily be distributed and run on thousands of machines around the world. Once this piece of code starts to perform, its effects might reach a scale of unaccountable proportions. It is as if code became possessed with magic powers which allow it to act upon us no matter where we might be.

## Code and the individual

So far I have only looked into a global scenario of code's performativity, however code also shapes our own subjectivity, that is, who we are as individuals. An interesting example is mentioned by Katherine Hayles (1999, p.47) in relation to virtual reality environments. In these virtual worlds, only certain stylized gestures can be encoded and have a manifestation, one has to move according to this limitations. By repeating this set of gestures frequently, they will surpass those moments in which one is hard-wired to the virtual, and will begin to alter our real bodies and behaviors. You may wonder why this example be considered to constitute a change of our subjectivity, of who we are. Perhaps it can, but not in a fundamental level. Matters seem to become more complicated when our communication, perception and experience of reality get mediated by code, then we are most certainly being changed by code. An example of this can be experienced in almost any rock gig. In this scenario the presence of many individuals is only validated if they bring home a piece of material evidence, such as a video on a mobile-phone, which proves that he or she had been there. One's own senses and memory are no longer valid ways of registering and reproducing reality. Information must be encoded digitally in order to become 'real'. A similar degree of code mediation goes for the ways in which we relate to each other. Social media, such as Facebook or dating websites, are among many examples of how this mediation is taking shape and molding our interpersonal relations.

Code is definitely shaping who we are and how we engage with our surroundings. However it is an extremely hard task to assess to what extent we are being changed. Since we are the matter that is being altered, it becomes very hard to pin-point what are the mechanisms in place and how are they shaping us. Perhaps that is the main reason why we accept such changes without much questioning. How can you question what you still do not understand?

## Conclusion

Looking back at what has been discussed, it seems relevant to reiterate that while code was given consent to act upon the world unmediated, our own experience of reality became heavily mediated by code. Reality itself must allow encryption into digital formats, making possible to store it in our digital archives, preventing it from vanish from the faulty human memory. As Hayles put it: "[d]ata are thus humanized and subjectivity is

computerized" (Hayles, 1999 p.39). Given these facts is not surprising that code has become such influential actor in our world.

Douglas Rushkoff in his book *Program or Be Programmed* envisions a near by future when computing is even more engrained in our society, which according to him will only give us two options: either write code or allow code to write us. According to Rushkoff computers gave us the possibility to write and make public what we write, never the less "the underlying capability of the computer era is actually programming" (Rushkoff, 2011, p.13) and that possibility is not being explored by most of us. Such delegation results in only a few of us being able to comprehend and influence the inner-workings of the bits of code that run our world. "Only by understanding the biases of the media through which we engage with the world can we differentiate between what we intend, and what the machines we're using intend of us - whether they or their programmers even know it"(Rushkoff, 2011, p.21).

I believe that learning to program is an essential skill in our times. It is a tool that not only allows us to get a better understanding of the many code based control mechanisms, but also allows us to be creative, to explore new venues and not be restricted to the tools which we are given, being them to create images, work with sound, communicate with our peers, collaborate, publicize our work, etc. It is with great sadness that I see schools and universities, despite all the investment put into giving each student a computer, still afraid to introduce their students to code, restricting it in most cases to engineering courses. Such delegation results in only a small percentage of the population taking the power and creative possibilities of code in their hands. It is perhaps time for all of us less technical folks to take code and begin to shape it for our own needs and desires.

## Bibliography

Cramer, Florian (2008). *Language*. In: Fuller, M. *Software Studies: a Lexicon*. London: The MIT Press. 168-174.

Cramer, Florian (2005). *Words Made Flesh: Code, Culture, Imagination*. Piet Zwart Institute

Galloway, Alexander (2004). *Protocol: How Control Exists After Decentralization*. Cambridge, MA: MIT-Press.

Hayle, Katherine (2005). *My Mother Was a Computer: Digital Subjects and Literary Texts*. London: The University of Chicago.

Hayles, Katherine (1999). *How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics*. London: The University of Chicago

Lynch, C.G. (2007). *Remembering Black Monday, When Computers Traded Too Many Stocks and Wall Street Crashed*. Available: [http://www.cio.com/article/147406/Remembering\\_Black\\_Monday\\_When\\_Computers\\_Traded\\_Too\\_Many\\_Stocks\\_and\\_Wall\\_Street\\_Crashed](http://www.cio.com/article/147406/Remembering_Black_Monday_When_Computers_Traded_Too_Many_Stocks_and_Wall_Street_Crashed). Last accessed 1st December 2011.

Medić, Darija. (2011) *Navigating in the age of magical reproduction*. MA. Piet Zwart Institute

McKenzie, Jon (2001). *Perform or Else: From Disciple to Performance*. New York: Routledge.

Rushkoff, Douglas (2010). *Program or Be Programmed: Ten Commands for a Digital Age*. New York: OR Books.

Retrieved from "[http://pzwart3.wdka.hro.nl/wiki/User:Andre\\_Castro/WritingResearch/essay01](http://pzwart3.wdka.hro.nl/wiki/User:Andre_Castro/WritingResearch/essay01)"

---

- This page was last modified on 15 January 2012, at 20:32.
- This page has been accessed 114 times.
  
- Privacy policy
- About NetworkedMedia
- Disclaimers