Jana Horáková

The Gestures That Software Culture Is Made Of

2016

https://doi.org/10.25969/mediarep/22230

Veröffentlichungsversion / published version
Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:
Horáková, Jana: The Gestures That Software Culture Is Made Of. In: *MAP - Media | Archive | Performance*. MAP #7 Media / Performance: On Gestures, Jg. 7 (2016), Nr. 1, S. 1–18. DOI: https://doi.org/10.25969/mediarep/22230.

Erstmalig hier erschienen / Initial publication here:
https://perfomap.de/map/7/media-performance-on-gestures/inverse-motion-of-thinking-on-technoscience-gesture-and-writing

The Gestures That Software Culture Is Made Of

Jana Horáková

To Program and Reprogram

The seductiveness of programming, gaming, and the processing of words, images, and sounds lies in the immense ease of manipulating the world within the computer. However, the ability to manipulate objects in a computer is not an expression of user freedom, but of a pre-programmed set of options, fonts, contrasts, and curving lines that program the user's activities.

The ability of information technologies to be programmed (and reprogrammed) and to program user activities and decisions has long escaped the attention of the media theorists. The advent of information technologies was accompanied mostly by visions of dematerialisation, disembodiment, and de-contextualisation, and thus of liberation from the constraints of the material world.[1] Since the beginning of the 21st century, more critical approaches within new media theory, in the sense of statements and arguments that have been more contextually and historically aware and anchored in the materiality of programmed media, can be detected, as well as more speculative and experimental methods of researching cultural production within information technologies.[2] Software studies is the most general name for these efforts to develop new and more appropriate methods of studying new media as programmed media (cf. Horáková 2011). Therefore, software studies will be used here as an umbrella term for these research methods.

Software Studies

The emergence of software studies implies an increased interest of media theoreticians in software in the sense of both the materiality of media (e.g. the code of programming languages, algorithms, and programmability) and its cultural dimensions: the ways the software, its logic, and the processes that take part within and through programmed media participate in different fields of production, storage, and dissemination of information, from the most creative or radical (computer viruses or software art) to the most ordinary and omnipresent (office work). Matthew Fuller expressed a demand for

broader and more context-oriented investigation of programme media in his call for (software) studies that

would fold 'back into software in its existence as culture.' (Fuller 2008: 6)
Software studies are defined as a new research field combining the viewpoints of scholars and programmers in order to offer a contrast to mere utilitarian and goal-oriented approaches of computer scientists. Software studies offer a broad view of computer programming and its results – software objects and processes. Computer programs (software) are understood as cultural products with a rich history that goes beyond that of the modern computer, as these objects and processes intrude on varied cultural productions that are significantly influenced by the ethos of the subcultures surrounding new media – from hackers and activists proposing free and open software to artists experimenting with code and programming in unexpected ways, thus testing the limits of programmability, virtuality, and automatization of work in post-industrial production. (Fuller 2008: 6)

The need to transform media studies (elaborated upon the theory of communication, approaching new media production from the viewpoint of the user, and interpreting the effects of the computer work that takes place on the surfaces of the screens only) to software studies (media studies informed by the logic of new media and approaching these media from the viewpoint of the programmers) was declared by Lev Manovich in 2001 for the first time:

> To understand the logic of new media we need to turn to computer science. It is there that we may expect to find the new terms, categories, and operations that characterize media that became programmable. From media studies, we move to something that can be called 'software studies'— from media theory to software theory. (Manovich 2001: 65)

A few years later, Matthew Fuller developed Manovich's idea of software studies into a concept of a new interdisciplinary research field within cultural studies that would enable more sophisticated research of contemporary 'software culture':

> As software becomes a putatively mature part of societal formations, or at least enters a phase where generations are now born into it as an infrastructural element of daily life and specialist practice, as such it gathers and makes palpable a whole range of associations, interpretative frameworks, qualitative dimensions of relationality, aesthetic and political bottlenecks and amplifiers and logico-cultural dimensions and aporias. (Fuller 2006)

Within software studies, software is understood both as a cultural object and as a powerful agent with immediate social and cultural influence. It is studied in its various

forms, material and immaterial, conceptual, and functioning within hardware as well as within different cultural practices. Information technologies are not seen as mere media that transport messages from point A to point B. They are defined as programmed media working in real time and are characterized by their performativity (cf. Arns 2004). They are seen as part of a complex environment, a network of relationships in which negotiations on centres of power, control, and meaning are taking place.

An Apparatus: The Software Studies Playground

Software studies has a lot in common with Vilém Flusser's philosophy of media concerning the concepts of apparatus,[3] functionary,[4] and processes inscribed in programs[5] that take place through and within apparatus-functionary arrangements.

Flusser is generally acknowledged as a theoretician of photography. His writings can be found in almost every anthology of classic writing on photography, next to Walter Benjamin, Susan Sontag, and Roland Barthes. However, his writings differ from these theoreticians in one important aspect. Flusser studied photography (camerawork) as an example of the work of an apparatus; the camera is one kind of apparatus for him. He understood the camera and video recorder as media in transition from tools, through machines, to computers (information-processing technologies). It is the computer in which all the features and tendencies that he recognised in apparatuses of previous stages reached fulfilment, and thus they are manifested in its completeness (see Flusser 2003). Flusser's writings on the camera should therefore be read with this perspective, as contributions to the genealogy of information technologies defined by their programmability, and his theoretical work should be retrospectively involved in the corpus of software studies.

Apparatuses are, according to Flusser, cultural products, or 'informed objects'. By studying the apparatuses, we can explore the culture that produces them (Flusser 2003: 17). Flusser defined an apparatus as …

> [A] complex plaything, so complex that those playing with it are not able to get to the bottom of it; its game consists of combinations of the symbols contained within its program; at the same time this program was installed by a metaprogram and the game results in further programs; whereas fully automated apparatuses can do without human intervention, many apparatuses require the human being as a player and a functionary. Apparatuses were invented to simulate specific thought processes. Only now (following the invention of the computer), and as it were with hindsight, it is becoming clear what kind of thought processes we are dealing with in the case of all apparatuses. That is:

> thinking expressed in numbers. All apparatuses (not just computers) are calculating machines and in this sense 'artificial intelligences', […]. In all apparatuses […], thinking in numbers overrides linear, historical thinking. […] [the apparatus] is computational thinking flowing into hardware. Hence the quantum (computational) structure of all the movements and functions of the apparatus. In short: Apparatuses are black boxes that simulate thinking in the sense of a combinatory game using number-like symbols; at the same time, they mechanize this thinking in such a way that, in future, human beings will become less and less competent to deal with it […]. (Flusser 1983: 31/32)

From ornament to fractal

Flusser was aware of the complex structure that apparatuses establish through their synchronised and mutually interconnected activities. He used photography as an example with which he was able to describe the complexity of the apparatus-functionary system. For him, to take a photo did not mean to only follow the program of photography implemented in the camera. It meant to take part within the complex infrastructure of the functioning of the programs:

> […] that of the photographic industry that programmed the camera; that of the industrial complex that programmed the photographic industry; that of the socio-economic system that programmed the industrial complex; and so on. (Flusser 1983: 29)

The hierarchy of programs listed by Flusser is open in both directions. Thus, programmers of any program are simultaneously functionaries of the metaprogram. Siegfried Kracauer described an industrial society apparatus using the metaphor of a 'mass ornament' (Kracauer 1995); Flusser added another dimensions and dynamics to the model to be able to describe the meta-apparatus character of post-industrial society. Following the functioning of the apparatus made of both apparatuses of the industrial age and apparatuses of the post-industrial age, Flusser found that the shape of the ornament at all levels and at all scales, macro and micro, is the same (Flusser 2013: 75-82). Moreover, the maze of the post-industrial apparatus expands in all directions, towards gigantism and miniaturisation, and its borders are disappearing from our view beyond the horizon while they concurrently become imperceptibly small (as chips, atoms, and genetic information). 'We swell-up and shrink, simultaneously,' wrote Flusser (Flusser 2013: 75). Thus, it can be said that the model of post-industrial apparatus does not take the shape of a two-dimensional ornament of industrial society but becomes a multidimensional fractal-like setting. In Matthew Fuller's words:

> […], iterations of multiscalar relations of causality and interpenetration are compiled layer upon layer. Base and superstructure shot through a kaleidoscope. […] The relation is simply one of scale, of order. (Fuller [no year]: 2)

There is no place outside the fractal ornament of post-industrial production. The programmed forms of post-industrial apparatus are elastic traps, invisible and comfortable for many (Flusser 2013: 75–78).

> Beyond the apparatus, there is nothing to do. […] Only thanks to cybernetic apparatuses, 'creation' is a concept quantifiable by a means of information theory. […] Where apparatuses prevail, there is nothing left to do but function. (Flusser 2014: 17)

Fuller appreciates the fuzziness and complexity of Flusser's concept of the post-industrial apparatus described as the infrastructure of 'multiscalar relations of causality and interpenetration […] [in which] programs and metaprograms are never clearly defined as distinct.' (Fuller [no year]: 2)

The superstructure of 'the apparatus of apparatuses' can be taken as a point of departure for the new research strategies proposed by Matthew Fuller under the names of software studies, media ecologies, or evil media. He drew attention to Flusser's notion of technology as a bearer of forces and drives that can be 'technical, aesthetic, economic, or chemical' (Fuller [no year]: 2) that corresponds with his own proposition to study programmed media as discursive objects.

Another influential legacy of Flusser's media theory lies in the method rooted in the phenomenological style of thought. His definition of the apparatus resulted from careful observation of the processes taking place through and around apparatuses. This enabled him to transcribe the traces of the processes (and thus also transcribe the forces and dynamics that the processes carry on or are animated by) that pass through all kinds of bodies (human beings, bodies of data, or bodies of machines) and thus constitute the assemblage of the 'super-apparatus' in his writings. This methodological strategy enabled Flusser to reveal that the activities of the operator are part of the program that the apparatus follows. In other words, the operator

> […] doesn't work anymore, but rather functions as the functionary of a function. This absurd gesture cannot be grasped without observing machines, for we are actually functioning as functions of machines, which function as the function of a functionary, who in turn functions as the function of an apparatus which functions as a function of itself. (Flusser 2014: 13–14)

The metabolism of an apparatus

The description of the post-industrial apparatus is reminiscent of H. R. Giger's bio-mechanic surrealistic paintings, in which grotesquely intertwined bodies and machines lose the character of closed, autonomous entities in favour of representing processes in the form of ornaments of metabolism that take part within the human-machine arrangement. Stelarc, a performer exploring the post-industrial state of the evolution of the human species, properly depicted the post-industrial apparatus in a collage of phrases written in theoretical and poetic language describing his performance *Fractal Flesh*:

> A body of FRACTAL FLESH, whose agency can be electronically extruded on the Net—from one body to another body elsewhere. Not as a kind of remote-control cyber-Voodoo, but as the DISPLACING OF MOTIONS from one Net-connected physical body to another. Agency could be shared in the one body or in a multiplicity of bodies in an ELECTRONIC SPACE OF DISTRIBUTED INTELLIGENCE, a body with TELEMATIC SCALING OF THE SENSES, perceiving and operating beyond its biology and the local space and human scale it now occupies, a body remapped and reconfigured, a body directly wired into the Net, a body that manifests the statistical and collective data flow, as a socio-neural compression algorithm. A body whose proprioception responds not to its internal nervous system but to the external stimulation of globally connected computer networks.[6]

Flusser's description of the functioning of the apparatus implicitly unmasks the illusiveness of the concept of an autonomous subject. In his depiction of the apparatus, the functionary (the human) is defined by the interaction with the surroundings in which it is immersed. The body as a closed object (black box) is replaced by a model of super-apparatus metabolism, in which all its parts are interconnected in a food chain, where everyone is consumer and producer, predator and prey, subject and object of action or thought, depending on the point of view.

Tactics of Freedom Within the Apparatus

Flusser called for new appropriate tactics for work, play, and thought (inseparable within the post-industrial apparatus), that would be able to cut 'the magical circle' of human-machine functioning within the apparatus, and establish a new condition of a self-aware, reflective style of thinking in the space of the rupture. Fuller developed Flusser's concept further by searching for an appropriate way to question the nature of the apparatus and the purposes and intentions programmed into it and emerging

within it, and thus to develop a certain body of knowledge about the character of the arrangement.

Fuller starts by pointing towards the place/position of the theoretician in relation to the object of their research – the post-industrial apparatus made of apparatus-functionary relations. He reminds us that if we accept the omnipresence of the post-industrial apparatus, the concept of thinking as an autonomous process is untenable. There can be situated, and thus political, knowledge only. Thought must be understood as a dynamic process, one 'that establishes another dimensionality to a problematic, in this case apparatus-functionary interactions within the post-industrial apparatus, rather than separate from it.' (Fuller [no year]: 28)

With reference to Foucault's definition of thought as 'freedom in relation to what one does, the motion by which one detaches oneself from it, establishes it as an object, and reflects on it as a problem,'[7] Fuller suggested that 'thought is a freedom' (Fuller [no year]: 28) and that in this respect the theoretical work follows the same strategy as art does (Fuller [no year]: 27). He named two main tactics of freedom, which can be called 'self-reference' (cf. Fuller 2006) and remediation (Bolter and Grusin 1998) (or scripted media [Weibel 2009]), and which have been developed in art but can be effectively applied in theoretical work as well.

> Taking a rule, a logic, a procedure and turning it over upon itself, or applying it in another domain, is one of the revelatory and productive procedures to be developed further here. (Fuller [no year]: 27)

The theoretical work as an activity situated within the apparatus is defined then as a self-reference of both the machine (the intelligent and informed instrument) and the functionary. The self-reference then becomes one of the central strategies of research within the apparatus. As an example of the strategy of self-reference, Fuller uses a work by London-based artist John Hilliard, *Camera Recording its Own Condition (7 apertures, 10 speeds, 2 mirrors)*.

> A camera is positioned in front of a mirror. The photographer works through every combination of settings for aperture and speed. Each time the combination of settings is changed, a picture is taken. The hands working the outside of the apparatus appear along with the camera. They also hold a smaller mirror, showing the settings on the top of the apparatus. At the upper left of the grid of prints resulting from this program of work, the emulsion is left utterly unstained. From the bottom right corner, up to a diagonal margin, and encompassing almost a third of the prints, the results are completely black. Every size of aperture allowed by the camera is run through, correlated with a procedural workout of every shutter speed. As the various combinations are made, the two mirrors bounce the camera's own reflected light back to it and the film it contains is reorganised by a

> measured amount of exposure to light waves. The seventy negatives resulting from this procedure allow, through the prints derived from their transition from whiteness to blackness, the gradually appearing and disappearing image of the camera to be seen. The program of the image is both built and erased by the apparatus which composes it. In 'photography', the proper use of a camera, the degree of darkness or lightness of the image indicates the closeness to or distance from sources of diffuse or direct light of the object being photographed. Such use of the apparatus is here revealed as being precariously lodged between whiteout and blackout. (Fuller [no year]: 4)

What is of interest to Fuller in this artwork is the 'staged … particular set of programs (speed and aperture) embedded within the camera…' (Fuller [no year]: 30) 'The images are the result of the self-referential performance of the apparatus. They 'articulate the camera as the possessor of compositional drive – form is generated by the material qualities of the camera, light, mirror, film, apparatus and the program which compose them.' (Fuller [no year]: 30)

This camera work of art illustrates well the power of self-reference as a thought strategy that can be applied in both art and theory. Self-reference does not apply only to images. To show the universality of the strategy, Fuller uses the example of feedback in music, in the situation in which

> sound produced by bringing the microphone, guitar, or other element of sound into such a relation of proximity with its output – a speaker – that the sound of the speaker itself becomes input, begins to vibrate the diaphragm of the mic for instance. The system becomes cyclical and positive – it begins to amplify its own amplification. (Fuller [no year]: 31)

The example of the self-reference of the sound system shows how a slight change in distance can make a big difference that results in the amplification of the system/apparatus function, and thus makes it perceivable and thinkable. By slightly altering the standard arrangement of the elements connected in a cybernetic feedback loop, the limits of programming, calculation, occurrence, or the system can be made perceivable.

Software art and tactics of self-reference

Software studies builds upon the cultural history of computing, hacker culture vernacular theory, and the ethos of free software, but the most direct and most important inspiration comes from software art.[8] Fuller argued that

> This current of work, […], provides a means for bringing the generative, reflexive, and anarchist intelligence of art into compositional entanglement with the ostensibly ordered and self-sufficiently technical language, working patterns, and material of software. (Fuller 2008: 8)

Software art emerged within the net art scene in the 1990s, and is distinctly different from computer art of the 1960s and 1970s. Artists from the software art scene criticise the pioneers of computer art for presenting the computer as a medium of elusion and abolition of intentionality, and as a means of producing arbitrary and endless diversity that occurs as if outside of human culture (Arns 2004). Computer art did not provoke the audience to ask questions about the nature of the processes that result in the appearance of images and texts on screens or in the production of sounds on computers. Software art, with its strong investigative, speculative, and critical ethos, is therefore not considered to be a successor to computer art. Even media theoreticians prefer to situate the new, post-1990 media art production within other experimental art forms of the 20th century; however, they do not even mention computer art as an early stage in the new media art production of the late 20th century (Tribe et al. 2006).

Software art can be understood as an exploration of the limits of freedom within a set of rules and opportunities that the post-industrial apparatus offers. Thus, it can be described as a programming of excesses in the sense of movements across the boundaries of the rules, functions, and limits inscribed into software and computer hardware.

The goal of software art is to investigate software, meaning the formal instructions and their execution, which are the essence of programmed media. The artists often work in the border zone of programming, since it is the limit of the phenomenon that enables the inspection of its properties in the form of pure functionality of the principles that it embodies. The following examples of software art combine self-reference, the programming of excesses, and the strategies of investigating the limits of programmability and thus the limits of the post-industrial apparatus.

Every Icon

Every Icon is a classic work from 1996/97 by John F. Simon, Jr., a member of the committee for the software art category at the *transmediale* festival in Berlin 2001. *Every Icon* was inspired by Paul Klee, who had written a note about a plan to explore the possibilities of combinatorial patterns and structures by filling in the squares of grid drawn on paper in his diary (Tribe et al. 2006: 86). Simon took Klee's idea and transcribed it into a simple program that can be run on a web browser:

Given: A 32 X 32 Grid
Allowed: Any element of the grid
to be black or white
Shown: Every Icon

Simon explained that

> Every Icon progresses by counting. Starting with an image where every grid element is white, the software displays combinations of black and white elements, proceeding toward an image where every element is black. In contrast to presenting a single image as an intentional sign, Every Icon presents all possibilities.[9]

The total amount of black and white squares in a grid of 32 x 32 is 1.8 x 10. This means that at a speed of 100 squares per second, which is the average speed of a personal computer, it will take 1.36 years until all of the variants of the first grid line will have been displayed. Exhausting all of the alternative combinations with a second row will take exponentially longer — 5.85 billion years to complete.

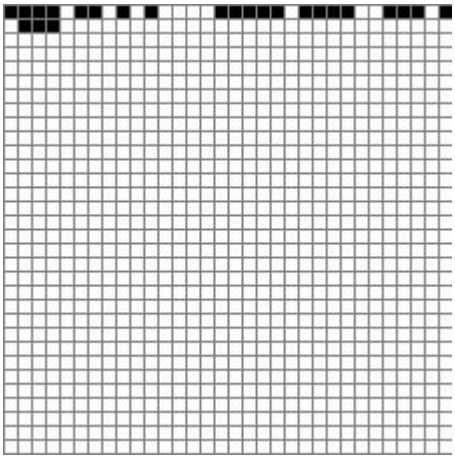

Figure 1: John F. Simon Jr. Every Icon, 1997.
http://www.numeral.com/eicon.html

The author rejected choosing some of the generated images on the basis of aesthetic judgement. He decided to present all of the options that the 32 x 32 grid of black and white would bear. He wrote that 'In contrast to presenting a single image as an intentional sign, *Every Icon* presents all possibilities.' (Simon 1996)

Every Icon is a program that, due to the durability of human products, will never be fully completed. The work can be appreciated conceptually, but its extreme length does

not allow us to see it fully accomplished in physical reality. So it is not only an example of programming of excess but also an example of conceptual programming, considering the limits of the programmed media.

Self.pl

A work by artist and media theorist Florian Cramer, entitled *Self.pl.* (2001), is a minimalist artistic score, written in the Perl programming language that is a functioning computer program and a poem at the same time.

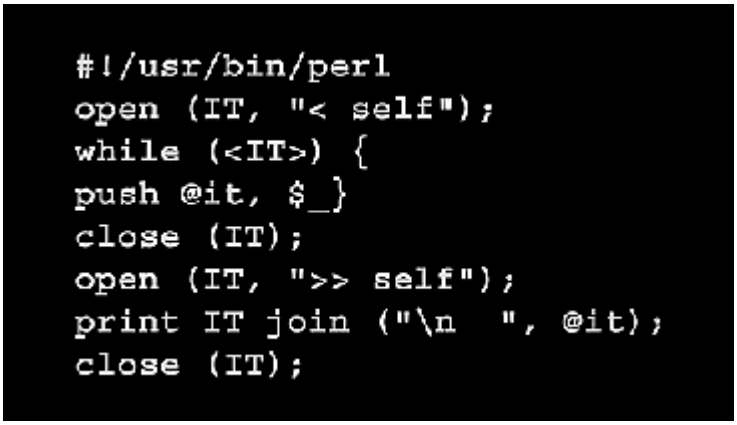

Figure 2: Florian Cramer: Self.pl, 2001.
http://www.uweb.ucsb.edu/~mweismann/art7d/software.html

The algorithm of *Self.pl* can be expressed in natural English as follows: 'open it self while it push at it close it open it self print it join at it close it.' (Horst 2008: 105) This software poem can be read quite easily, because it contains a number of elements of natural language. But it is a command line addressed to the computer at the same time. When the particular program runs on a computer, it produces an empty folder named 'self'; then, the program reads the content of the folder, which is empty, closes and reopens it to write down what it just has read – which is nothing. Finally, it prints out a blank page.

This software work of art can be read in three ways: as a poem written in English, as a sequence of computer instructions, and, in the moment it is processed by the computer that is printing out the text, as a new poem in English (Horst 2008: 103). It is thus not just a poetic text; the performance of self-reflexivity of the code reveals the performative character of computer programs that are scripts and performances at the same time.

Self.pl can be interpreted as a literary self-portrait, a performance of 'self' that writes itself. Thus, it is a classic example of a self-reflective work of art. At the same time, it is a minimalistic program, which by its simplicity and poetry contrasts both with the sophisticated applications typical of early virtual reality art, and with commercial desktop applications hiding their coded natures behind a pictorial interface.

Forkbomb

Hackers use the word 'fork bomb' for programs that work on the principle of branching, when one running process generates the launch of other processes. The gradual launching of processes is so fast that it quickly leads to the overload and freezing of the computer operating system. Ordinary users experience something similar in situations when they run too many applications at once on their computer, and the computer lags. A 'fork bomb' is one of the most popular genres of hacking production, focusing on writing elegant and simple but effective codes, and it was very popular in the mid-1990s. In 2002, one such 'software bomb' won an award in software art at the *transmediale.02* festival in Berlin.[10] The award-winning piece was *Forkbomb* (2001/2005) by Alex McLean, written in the Perl programming language.

The output of the *forkbomb.pl* program, which is about thirty lines long, are binary data, zeros and ones arranged as if randomly on a black background, creating the impression of a kind of minimalist music score. The form of the output on the monitor is determined by an algorithm represented in code and partly also by the operating system on which the program is running. The computer operating system is in a state of constant change; therefore, the performance of this script will always produce different results. McLean interprets these outputs as an expression of an artistic impression of the system in a state of overload.

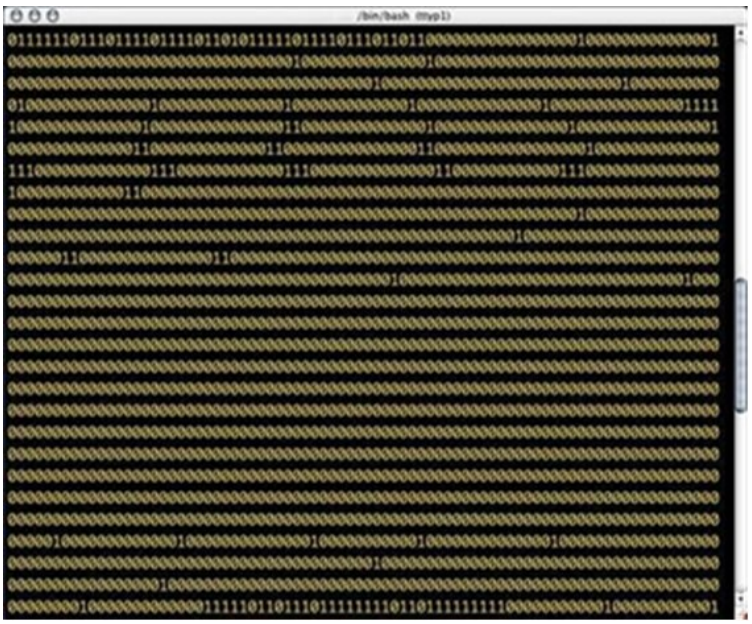

Figure 3: Alex McLean: Forkbomb.pl, 2001/2005.
http://www.medienkunstnetz.de/werke/forkbomb/

Software Art, Software Studies, and the General Theory of Gesture as Gestures Within the Apparatus

Flusser introduced a rather ambivalent vision of human existence within the post-industrial apparatus. He claimed that post-industrial apparatuses (programmed intelligent machines) have a tendency to transform any environment they are put into to the apparatus, and that the post-industrial society is fully equipped with apparatuses. The terms 'post-industrial society' and the 'post-industrial apparatus' can then be used as equivalents.

Following the algorithm of the argument that each apparatus is driven by programs stored in it, and that these programs define all possible activities of the functionary as well, Flusser recognised two tendencies of human existence within the apparatus – towards the robotisation of humans and towards liberation, in the sense of human emancipation from work for creativity (play), even within limits defined by the apparatus. Software art projects, including those introduced above, can be seen as manifestations of human creativity, and thus of their liberation within the post-industrial apparatus.

Software art is defined as a creative investigation of the limits of programmability of the post-industrial apparatus by means of the programming of excesses or self-

referential software. Software art can be seen as a part of a performative turn in art that marks increased artist interest in categories such as process, movement, duration, environment, and audience participation, as well as in understanding artwork as a process or an event; in other words, as a symptom of an artistic gesture, in contrast to the traditional object-oriented definition of artworks as autonomous artefacts.

The performative turn has also influenced academic discourse. Scholars across different disciplines have started to investigate human activities such as utterances or gestures. Within the non-representational theory that the performative turn brought about, utterances and gestures are understood not as manifestations of codes and symbols, but as recurring manifestations of different modes of knowledge. Software studies, with their focus on the processuality and performativity of programmed media, can be understood as a part of the performative turn, expanding the research area beyond human activities towards hybrid processes that cross the human-machine arrangements of the post-industrial apparatus.

Flusser's proposal to establish a new 'theory of interpretation of gestures' (Flusser 2014: 2), called the general theory of gestures, which would evolve within a new field called 'communication research', can also be seen as a part of the performative turn in media theory. The new discipline is introduced as a method of searching for symptoms of freedom – the gestures, in Flusser's words, within the telematic embrace of the post-industrial apparatus. The method is based on a distinction between movement and gesture.

> The concept of gesture may be defined as a movement that expresses a freedom. The gesture, as a kind of movement, is in fact determined, as are all other movements, and in this sense completely explainable. But what makes it unique, distinct from other movements, is that it expresses a subjectivity … that we are forced to call freedom. (Flusser 2014: 173)

Flusser introduces the theory of gesture as a potentially revolutionary way of reflecting on the ways we are within the apparatus – and on the ways we could be. He explains it as 'a discipline for a future' and 'a discipline of so-called new-people' (Flusser 2014: 176).

> This sums up the engagement of the theory presented here: to contribute to an enhanced freedom, and to be able to actually make gestures in the full sense of the concept defined previously. … Such a theory would not be value free; rather, its value would be freedom. It would consciously be an instrument of liberation and so anti academic. (Flusser 2014: 175-176)

The general theory of gesture can be seen as an older sibling to software studies, because both can be seen as a complementary theory of software art, the artistic

investigation of the limits of programmability, subjectivity, politics, ethics, and aesthetics within programmed media. Software art can be called the art of gestures or the art of freedom within programmed media (the post-industrial apparatus) in reference to an autonomy both of the machine and of its functionary.

In his theory of gesture, Flusser puts at the centre of art-production and (human) sciences the embodied subject interpreting the world (in contrast to a mere explanation of the word, a method typical of hard sciences). It can be said that the theory of gesture as a method based on searching for gestures within movements is itself a kind of gesture; it is a freedom manifested in gestures of interpretation. The typically human ability to interpret might be the only way to make sense of the absurd combinatorics game of the apparatus that we are part of.

Conclusion

The paper introduced Vilém Flusser's concept of a post-industrial apparatus as a predecessor of Matthew Fuller's concept of software culture situated within the programmed media network. Both authors dealt with the programmability of the apparatus/media as a crucial feature that radically transforms the human-machine relationship in comparison to the ways that humans use tools. Flusser and Fuller share a certain anti-academism that manifests itself in blending theoretical work with interpretations of hands-on experiences of artists, programmers, or interpreters. Another shared feature is their interest in research methods, as manifested in their search for new methodologies appropriate for post-industrial apparatus or software culture research (the general theory of gesture, software studies, and others).

Fuller finds inspiration for tactical thinking about programmed media in (software) art that investigates the limits of programmability and reveals the character of the new media by staging close loops of media self-reference. Flusser proposes developing a new theory of interpretation of gestures (expressions of freedom), rather than mere functioning (movements) within the apparatus. Both authors transcend the theoretical discourse towards engaged, critical writings, in order to increase the awareness of functionaries/users of programmed media about their abilities to program their behaviour and thought. However, they do not stop with the mere critique of the situation; they indicate tactics for revolt against the program of apparatus. Flusser calls for the search for gestures of freedom, and Fuller tests the possibilities of appropriating software art or hacker culture strategies within software culture theory. Both search for an adequate new human science that would offer an appropriate method of searching for gestures of freedom within the apparatus.

Jana Horáková (born 1971) is an associate professor of Theory of interactive media at the Department of Musicology of the Faculty of Arts of Masaryk University, Brno. She studied theatre studies at Charles University in Prague and Masaryk University in Brno and media studies at the University of Lapland, Finland. She focuses on media art and performance art, historical and theoretical interrelations and on robotic art (Book:*Robot as Robot*, KLP: Prague 2011). She met with Barbara Büscher on the first Media-Performance symposium (Goethe-Institute, Prague 2005) and since that time they have continuously cooperated. They organized the Czech-German symposium Media-Performance 2 on ´ephemerality´ (Dům pánů z Kunštátu, Brno 2007), Media-Performance 3 on ´memory´, and they are co-editors of Czech-German publication: *Imaginary Spaces: Raum/Prostor - Medien /Média - Performance/Performance* (KLP: Prague 2008). In her latest publication *Software Studies.Towards New Media Studies Transformations* (2014) she introduces main concepts and milestones of software studies that she interprets as significant methodological turn.
e-mail: horakova@phil.muni.cz

1 For more about the stages of media studies, see for example Hui Kyong 2011.

2 These can be found on the internet using key words: software studies, critical code studies, platform studies, evil media studies, etc.. See also Horáková 2014 (in Czech with an English summary).

3 'Apparatus (pl. -es): a plaything or game that simulates thought [*trans*. An overarching term for a non-human agency, e.g. the camera, the computer and the 'apparatus' of the State or of the market]; organization or system that enables something to function.' (Flusser 1983: 83)

4 'Functionary: a person who plays with apparatus and acts as a function of apparatus.' (Flusser 1983: 83)

5 'Program: a combination game with clear and distinct elements [*trans*. A term whose associations include computer programs, hence the US spelling].' (Flusser 1983: 84)

6 A collage of phrases and theoretical and poetic language for the performance *Fractal Flesh*, published on his website: Source: http://www.stelarc.va.com.au/fractal/index.html Taken from: http://www.medienkunstnetz.de/works/fractal-flesh/

7 Fuller [no year]: 28 (quote no. 61: Foucault 1997, p. 117).

8 see the RunMe.org repository of software art, available at http: // www.runme.org, and the catalogues for the Read_Me festivals, edited by Olga Goriunova and Alexei Shulgin.

9 *Every Icon* (1996) is a project by John F. Simon, Jr., Artist's statement: http://www.numeral.com/articles/paraicon/paraicon.html

10 McLean's *forkbomb.pl* is a variation of a famous computer science exercise, which suffocates the computer with its own processes. *forkbomb.pl* won an award as a crafted code which was transformed into an artwork of technical transgression. http://tekspost.no/pipermail/syndicate/2002-February/004608.html

References

Arns, Inke. "Read_me, run_me, execute_me. Code as Executable Text: Software Art and its Focus on Program Code as Performative Text." In: Tjark Ihmels and Julia Riedel (eds.), *Generative Tools*. 2004. http://www.medienkunstnetz.de/themes/generative-tools/read_me/scroll/ (retrieved 12 February 2016)

Bolter, Jay David and Richard Grusin. *Remediation. Understanding of Media*. 1998.

Flusser, Vilém. "Beyond Machines." In: *Gestures*. University of Minnesota Press 2014.

Flusser, Vilém. "Gestures and Affect." In: *Gestures*. University of Minnesota Press 2014.

Flusser, Vilém. "Our Shrinking." In: Siegfried Zielinski (ed.), *Post-History*. Univocal 2013.

Flusser, Vilém. "Towards a General Theory of Gestures." In: *Gestures*. University of Minnesota Press 2014.

Flusser, Vilém. *Krise der Linearität*. (The text of Flusser's speech in Bern: Benteli 1988. Flusser Archive, Berlin.) In: Nils Röller and Silvia Wagnermaier (eds.): *Absolut Vilém Flusser*. Freiburg: Orange-Press 2003.

Flusser, Vilém. *Towards a Philosophy of Photography*. Reaktion Books 1983.

Foucault, Michel. "Polemics and Problematization. Interview conducted by Paul Rabinow." In: Paul Rabinow (ed.), *The Essential Works of Michel Foucault 1954-1984 Vol. 1: Ethics, Subjectivity and Truth*. New York: The New Press 1997.

Fuller, Mathew. *Software Studies Workshop*. 2006. http://web.archive.org/web/20100327185154/http://pzwart.wdka.hro.nl/mdr/Seminars2/softstudworkshop.

Fuller, Mathew. *Software Studies: A Lexicon*. MIT Press 2008.

Fuller, Matthew. "The Camera That Ate Itself." In: *Flusser Studies 04* (no year). http://www.flusserstudies.net/sites/www.flusserstudies.net/files/media/attachments/fuller_the_camera.pdf

Fuller, Matthew. *Softness, Interrogability, General Intellect: Art Methodologies in Software*. Huddersfield University 2006.

Horáková, Jana. "Konec dějin nových digitálních médií: Softwarová studia." In: *Umění a nová média*. 1. vyd. Brno: Masarykova univerzita 2011, p. 156-179.

Horáková, Jana. Software Studies. Towards New Media Studies Transformations. Brno: Masaryk University 2014.

Horst, Phillip. "Code Poetry's Aesthetic WW(WEB). Examples of Contemporary Intermedial Experiments." In: Kornelia Freitag and Katharina Vester (eds.). *Another Language: Poetic Experiments in Britain and North America*. LIT Verlag 2008, p. 93-110.

Hui Kyong Chun, Wendy. *Programmed Visions: Software and Memory*. MIT Press 2011.

Kracauer, Siegfried. *The Mass Ornament. Weimar Essays*. Edited and translated by Thomas Y. Levin. Harvard University Press 1995.

Manovich, Lev. *The Language of New Media*. MIT Press 2001.

Simon Jr., John: *Every Icon*. 1996 http://www.numeral.com/articles/paraicon/paraicon.html (last visited 12 February 2016).

Tribe, Mark, Reena Jana, and Uta Grosenick: *New Media Art*. Köln: Taschen 2006.

Weibel, Peter. "Synthetic Times." In: Fan Di'an and Zhang Ga (eds.), *Synthetic Times*. Cambridge, MA and London: MIT 2009, p. 112–142.