

Sub disorder() after 'The Author as Producer'

'An author who has carefully thought about the conditions of production today [...] will never be concerned with the products alone, but always, at the same time, with the means of production. In other words, his [/her] products must possess an organising function besides and before their character as finished works.'
(Walter Benjamin, 'The Author as Producer', 1992 [1934], p.98).

This essay asks if Benjamin's line of thinking retains relevance for cultural production at this point in time,¹ when activities of production, consumption and circulation operate through complex global networks served by information technologies. The dynamic and algorithmic routines of the networked computer are now crucial to the production process in general, and it follows that critical attention should be given to its inner workings and the software on which it relies. An underlying assumption of the essay therefore is that software is not simply a functional tool but expresses wider cultural and technological processes that extend the critical potential of arts practice. Its appreciation requires this simultaneous understanding of the production of its source code and its execution. This sounds obvious perhaps, but it is quite common practice for examples of software art to hide the complex interactions of processes and code running on the computer behind the scenes, as well as the working processes of both programming and of programmers.

With some of these ideas in mind, there has been considerable recent attention to software art and culture, and this is partly due to a range of cultural events that focus attention on the social implications of programming and the materiality of code. For instance, to mention some of the key examples: the *Readme* festival and its associated *Runme* software art repository in Moscow, Helsinki, Aarhus, Dortmund (2002-4);² the yearly *transmediale* media arts festival in Berlin that included a software art category until recently (2001-2004);³ and the *CODEDOC* exhibition on the Whitney Museum's *artport* (2002), and later at Ars Electronica (2003),⁴ in which the viewers of the work were invited to first read the source code and then see the executed work. In all these cases, software art is distinguished from practices in which software is used merely to generate an artwork. Instead, the emphasis is on software *as* the artwork - in keeping with Benjamin's attention to process over end-product expressed in the opening quotation.

On the surface, it seems that many contemporary artists operate in this spirit in keeping with Nicolas Bourriaud's claim that the image is now defined by its 'generative power', and that art can be seen to be a program(me) for the generation of

¹ Walter Benjamin, 'The Author as Producer', in *Understanding Brecht*, trans. Anna Bostock, (London: Verso, 1992). It was first presented as a lecture in April 1934 at the Institute for the Study of Fascism in Paris.

² *readme* (2002-) <http://readme.runme.org/> and *runme* (2003-) <http://www.runme.org/>

³ *transmediale*, <http://www.transmediale.de/>

⁴ *CODEDOC* (2002) online exhibition, <http://artport.whitney.org/commissions/codedoc/index.shtml>; and later *CODEDOC II* (2003) <http://www.aec.at/de/festival2003/programm/codedoc.asp>

forms and situations.⁵ He is referring to artwork that is a programme to be followed, a model to be reproduced, or an encouragement to do something - and points to the parallel activities of artists engaging in ideas of interaction and sociability set against the hype of interactive computer systems. But there is little new in his emphasis and there is a longer history of non-object based arts practices in which social relations are registered. For instance, in the 1970s, software as a metaphor for art was explored in Jack Burnham's exhibition *Software, Information Technology: Its Meaning for Art*,⁶ opposing the hardware or end-product of art. The importance of this reference lies with the centrality of transformation rather than representation, influenced by systems theory associated with Ludwig von Bertalanffy, and cybernetic theory associated with Norbert Wiener.⁷ In a sense, it can be said that Benjamin's call for a politics of representation (rather than the representation of politics)⁸ is upgraded to a politics of transformation.

What is radical about software remains as it did then, that it acts upon hardware and emphasises social relations between people, and people and machines. It operates as a metaphor for social processes and machinic agency. The term 'software' generally refers to a computer program and the resources related to it. In more detail, this means it includes not only the instructions written in a particular language as the program, but also the other material required for it to run, that are usually combined into software for distribution. Although it is important to stress the program itself as the source code that the computer executes, the general term software is useful as it describes the wider context within which the computer program runs. This contextual understanding also clarifies something about the use of the term 'software art', in describing a wider (operating) system of which it is part - emphasising that art conforms to formal structures and constraints but that these can be manipulated. For instance, Alex McLean's *forkbomb.pl* (2001),⁹ is a Perl script designed to take a computer to its operational limit. A computer under such high load causes unpredictable results that pattern differently depending on the operating system it runs upon. Its execution is particular to the apparatus or situation, rather than simply determined by rules - and so any tendency towards 'software-determinism' is rejected. Importantly, the example also stresses that other agencies are at work, including human agency as an integral part of the production process in which the rules are set. It demonstrates how the producer can concede control to some extent

⁵ Nicolas Bourriaud, *Relational Aesthetics*, trans. Simon Pleasance & Fronza Woods, (Dijon-Quetigny: Les Presses de Réel, 2002) p.70. To Bourriaud, artwork not using the computer has as much potential to make work about its effects.

⁶ The full title was *Software, Information Technology: Its Meaning for Art* at the Jewish Museum, New York, 1970. The exhibition was informed by Burnham's essays such as 'Systems Esthetics', in *Artforum*, vol. 7 no. 1 (September 1968).

⁷ An interest in cybernetics and ideas of feedback had already been tested in Jasia Reichardt's exhibition *Cybernetic Serendipity*, at the ICA, London (1968), which is widely regarded as a historical marker for first combining art and cybernetic ideas.

⁸ This is a reference to Benjamin's 'The Work of Art in the Age of Mechanical Reproduction', in *Illuminations* (London: Pimlico 1999 [1936 written]), pp.211-244.

⁹ McLean's *forkbomb.pl* was joint winner of the 'software art' award at *transmediale* in 2002. 'Perl' is an acronym for 'Practical Extraction and Report Language', a high-level programming language, first developed for Unix by Larry Wall in 1987, and developed as an open source project.

over the production of the work - and this is an important qualification - but that human intervention is fundamental to (software) production. In other words, the artwork is programmed - with or without the aid of a computer. Whether the artist was involved in the writing of the software or not directly is beside the point that someone was.

In terms of rule-making (the basis of programming), one might begin to differentiate between those who make rules, those that implement them, and those that are subjected to them. One can readily apply these formulations to the production of software (implicating the figure of the artist-programmer) and identify the subsequent relations of production. In other words, to demonstrate good 'technique', the artist-programmer must therefore combine the first and the last, on the behalf of the second - both conceive and implement rules as well as make put them in the public realm. With most commercial software the source code remains 'closed-off'. For the most part, the software compiles the code into an executable version that 'locks-down' the source to protect proprietary interests including the intellectual and artistic capital. That the source code might be considered an integral part of the artwork, or even the artwork itself, remains outside the imagination of the software/art market obsessed with property rights. This is where software art offers an alternative view perhaps, and can reveal contradictions over production such as these, peeling away the layers of operation and the relations of production involved in working with code, and in code working.

Class SoftwareArt

Inherits AuthorAsProducer

Sub disorder()

Dim A, B as Integer

Dim C as Color

A = Rnd * UBound(Pict)

B = Rnd * UBound(Pict)

C = Pict(A)

Pict(A) = Pict(B)

Pict(B) = C

End Sub

End Class

The title of this essay follows programming logic in presenting a 'subroutine' called disorder. As can be seen from the larger extract above, it is part of a larger project or 'subclass' (SoftwareArt) based on an existing work (AuthorAsProducer) that already exists as a 'class'. This logical sequence describes the purpose of this essay: it is about software art that adopts a general line of argument from Benjamin's essay and then argues for the transformative potential of disorder within systems. The subroutine is based on an extract from Ordure.org's *Dust* (2000),¹⁰ a dialectical image

¹⁰ *Dust* can be found at <http://dump.ordure.org/www.ordure.org/291/dust.html>. The source image was originally a photograph by Rosse Yael Sirb. The software was written by Adrian Ward, of The UK Museum of Ordure (<http://www.museum-ordure.org.uk/>).

that is both a representation and a process of detritus that slowly ‘corrupts’, pixel by pixel.¹¹ The corruption is triggered by viewing the image and in doing so a pixel moves from one location to another. The data is consistent, the pixels merely rearranged. Any comparison demonstrates a dialectical play between two interconnected states of order and disorder, between generation and corruption, suggesting the potential for transformation.

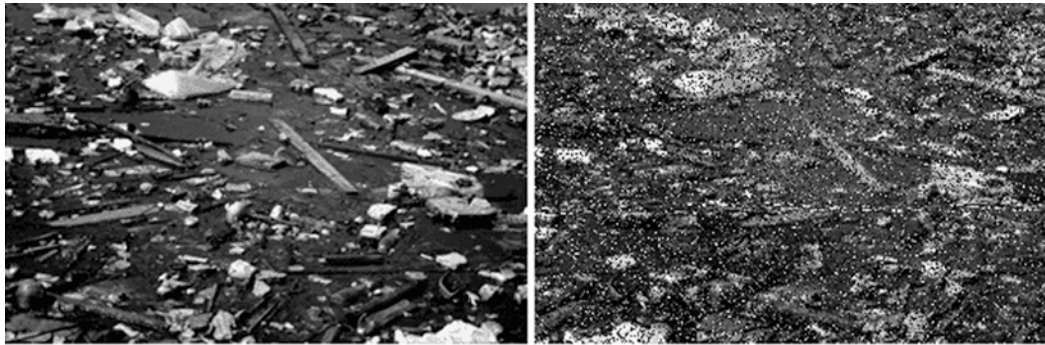


image caption: Ordure.org's *Dust* (2001)

Evoking systems theory, the suggestion is that systems are not closed but also open to influence and change from external and internal factors. The possibilities resulting from this are large and complex, but not endless nor open-ended as is often supposed. The relationship between order (that which can be classified and rationalised) and disorder (that cannot, because it is too chaotic and generalised) is a far more complex formation than simply an opposition. N. Katherine Hayles's contradictory phrase ‘orderly disorder’¹² suggests a more dialectical conjunction in keeping with the discovery that within the unpredictability of chaotic systems lie deep structures of order. So how does change come about? The system expresses unpredictability despite its deterministic character. That a relatively small input can have massive consequences presents analogies between complexity and social phenomena. Within self-organising systems, disorder can lead to order and in turn to new order. It is this that leads Wu Jie to suggest that just as the industrial revolution required the philosophy of dialectical materialism, so the new technological revolution requires what he calls *Systems Dialectics*¹³ - citing the parallel von Bertalanffy makes between the general systems principles and dialectical materialism. Both systems theory and dialectics share a common interest in conceptualising the material world in terms of processes, as the interaction between parts of the system and the development of the whole system that expresses dynamic interactions and contradiction. Wu explains his thinking:

‘This upgrades the law of negation of negation to the domain of the disordered - ordered - newly disordered - newly ordered, which is a more extensive and penetrating domain than the former.’¹⁴

¹¹ ‘Dialectical image’ is a reference to the dialectical approach of Benjamin in which he tries to halt the flow of the movement, as opposed to the more classical Marxist one where all social forms remain fluid.

¹² From her ‘Chaos as Orderly Disorder: Shifting Ground in Contemporary Literature and Science’, *New Literary History*, no. 20 (1989), p.305-22.

¹³ Wu Jie, *Systems Dialectics* (Beijing: Foreign Languages Press, 1996).

¹⁴ Wu Jie, *Systems Dialectics*, p. 88.

Like negation, new order might indeed be generated through disorder. According to this logic, at the point of antagonism, chance takes hold of determinism, and as a result either disorder or order may be generated. This theory of self-organising matter is also what Sue Owens has adopted in her essay 'Chaos Theory, Marxism and Literary History', wherein order is both expressed in chaos and might be generated out of chaos.¹⁵ It is the 'revolutionary' moment or 'bifurcation point' (where two paths are possible), and where dramatic change takes place but it remains impossible to predict the direction change will take. It will either fall into a higher level of order or disintegrate into chaos. This patterning continues in the spirit of dialectics that requires continual improvement so as to not stagnate and resist closure and a false sense of totality. Along these lines of thinking and in general terms, recent critical theory (exemplified by postmodernism) rests on 'bad science' and 'bad history' according to Owens (and we might add bad politics for this context). In this way, dialectical thinking combined with an understanding of complex systems challenges the pessimism of much contemporary critical thinking. Indeed one of the properties of complexity is that multiple interacting elements in a system cannot be 'governed' as such.¹⁶ Disorder thereby can be seen to release the social potential for transformation. Indeed, the system contains the seeds of its own destruction.

Despite conceptual problems with dialectics (associated with the position of Antonio Negri for instance),¹⁷ recent formulations continue to evoke dialectical formulations. For instance in *Empire*, contemporary power is described as 'characterised by a fluidity of form - an ebb and flow of formation and deformation, generation and degeneration'.¹⁸ In another formulation, the term 'corruption' is thought of in terms of 'de-generation - a reverse process of generation and composition, a moment of metamorphosis that potentially frees spaces for change'.¹⁹ The concept of corruption is especially evocative in this context as it lends itself to the description of destructive software forms that release the potential for further transformations – set against the existing corruption of the capitalist system. It follows that any characterisation of power must be countered with something equally complex or chaotic to be able to respond to the transformational mechanisms built into capitalism itself – as an adaptive system. Hardt and Negri describe this in the following terms: 'The Empire's institutional structure is like a software program that carries a virus along with it, so that it is continually modulating and corrupting the institutional forms around it.'²⁰

¹⁵ Sue Owens, 'Chaos Theory, Marxism and Literary History', in Jody Berland & Sarah Kember, eds., *Technoscience, New Formations*, no.29, Summer (London: Lawrence & Wishart, 1996), p. 86.

¹⁶ Evoked is the contradictory phrase 'governance without government', the title of a book by James Rosenau and Ernst-Otto Czempiel (1992), in Michael Hardt & Antonio Negri, *Empire* (Cambridge Mass.: Harvard University Press, 2000).

¹⁷ The problem Negri has with dialectics is bound up with Hegelian resolution and its implied attack of Spinoza's sense of 'immanence' which remains to the autonomists a revolutionary theory. Immanence in this sense, represents the potential against power, as an emergent force. It remains possible to read dialectics against this view.

¹⁸ Hardt & Negri, *Empire*, p.202.

¹⁹ Ibid., p.201

²⁰ Ibid., pp.197-8.

This is something Alain Joxe also responds to, in *Empire of Disorder*, in asking how resistance can be characterised to lead to a more pleasant chaos.²¹ His point is that traditional oppositional standpoints seem powerless to resist power, because it is so complex and has taken on forms of resistance itself. Order is now expressed through disorder in other words. This asymmetry between order and disorder is partly as a result of the ‘decomposition’ of bipolar cold war oppositions, replaced with the disorder of the free market (or ‘Empire’). As a result, what seems to be required is a response that draws on an understanding of complex systems. This is somewhat verified by Joxe's statement that:

‘Disorder is only a new beginning because it potentially contains a variety of possible orders, a variety of scales of possible orders. Disorder always opens a new choice of degrees of order.’²²

The question is whether traditional modes of resistance are suitable to resist power, because now power is more complex and has taken the form of resistance itself - new order is now expressed through disorder in other words. Central to this idea, is that change does not simply happen on a large scale socio-economic level or in ideology but increasingly from mutations at a micro scale molecular level, such as introduced by ‘microprocessors changes the actual substratum of human existence and, in reality, opens up fabulous possibilities for liberation’.²³ In other words, there is a dynamic tension between micro-politics and the body politic in general - integrating life and politics at all scales of operation. An important issue here is whether criticism operates most effectively from within or outside the system. Post-Marxists (Deleuze, etc.) criticise dialectics for this reason, as it can only ever operate within the very logic of the system it seeks to criticise. Surely this is its strength as it seeks to negate all that is irrational within the system. Is it really possible to operate outside of the system? A further problem arises in redefinitions of what constitutes the proletariat - partly to account for its failure to answer its revolutionary call. The term ‘proletariat’ as the subject of labour and revolt continues to be relevant but requires redefinition to stress more collective and communicative forms. But old antagonisms between capital and labour have not dissolved. On the contrary, they are enhanced and made complex. For instance, labour time is more difficult to measure and less distinct from time outside work, much of it now practiced as ‘nonwork’ outside of traditional production processes - *notworking* as opposed to *networking*. These tendencies correspond to the networked computer, in the way it has redefined social practices and relations, but the creative power to use technology differently, to reappropriate it, still rests with workers themselves as they are the ones who have the expertise to operate it. This expertise or ‘know-how’ is what Paolo Virno chooses to explore through a discussion of ‘virtuosity’ by looking at the special attributes of the performing artist. A performance is characterised by its lack of an end-product, or at least a product that is indistinguishable from the performance itself.²⁴ Furthermore, it operates in real-time

²¹ Alain Joxe, *Empire of Disorder*, (New York: Semiotext(e), 2002) p.107.

²² *Ibid.*, p121.

²³ Félix Guattari, *Chaosophy*, Sylvere Lotringer, ed., (New York: Semiotext(e), 1995), p.47-8.

²⁴ Paolo Virno, ‘Virtuosity and Revolution: The Political Theory of Exodus’, in Paolo Virno & Michael Hardt, eds., *Radical Thought in Italy* (Minneapolis: University of Minnesota Press, 1996), p. 52. Virno is drawing upon Hannah Arendt’s observation that the performing arts have a strong affinity to politics.

and has its own sense of purpose or fulfilment in parallel to the way that a computer program breaks down the distinction between its function as a score and its performance, or between code and its execution.

Systems operators, programmers, computer scientists, technicians, software engineers, software artists, office workers, and so on, clearly hold the potential to use and abuse this know-how. A dialectical approach associated with complexity theory would suggest that a refusal to work might be similarly implemented by disorder, by imagining alternatives and indeed producing software that invokes disorder. Clearly software is worked upon and performs work in itself, and as such some of the antagonisms associated with work can be made apparent and engaged. Software can be programmed to act in a relatively antagonistic manner, such as by the refusal to work (non-executable code) or by working in a negative mode or disorderly manner (dirty code), and outside the orthodoxy of passive work (analogous to proprietary models of clean and pure code). In the field of artistic software, these are familiar strategies, and the work of Jodi is particularly well-known. For instance, for *Documenta X* (1997), Jodi simply produced a link that on clicking made the visitor's machine crash.²⁵

For Benjamin, the writer-technician must reflect upon their position within the production process to demonstrate expertise alongside solidarity, acknowledging the choice of in whose service, or more particularly class interests, the producing operates. This, Benjamin explains, is usually called pursuing a tendency, or expressing 'commitment' and he takes this to be a key term.²⁶ In a similar way perhaps, McKenzie Wark's *The Hacker Manifesto* requires that hackers take control and seek autonomy over what they produce, to identify their interests as a class in order to serve society as a whole, and strike alliances with other workers who do not own the means of production.²⁷ Property disputes over software are revealing in this respect and support the view that new technologies necessarily challenge property rights. Wark maintains the emphasis on property, arguing that post-Marxist critique does not address the question of property sufficiently and the ways in which informational technologies have influenced the concept. His starting point is that Marx identifies that privatisation of the property relation creates class relations - firstly through land and subsequently through industrial capital. Intellectual property becomes a third, distinct form of private property, which gives rise to a third, distinct class antagonism. Wark explains:

'Just as the development of land as a productive resource creates the historical advances for its abstraction in the form of capital, so too does the development of capital provide the historical advances for the further abstraction of information, in the form of "intellectual property".'²⁸

Private property remains a cornerstone of capitalism. It remains, as it did at the time of Benjamin's writing, a key area of antagonism but needs some upgrading to stress

²⁵ Jodi's work can be found at <http://www.jodi.org>. In the UK, there was a touring exhibition of their work *Computing 101B*, commissioned by FACT, Liverpool (2004).

²⁶ Benjamin, 'The Author as Producer', p. 86.

²⁷ McKenzie Wark, *A Hacker Manifesto* (Cambridge, Mass.: Harvard University Press, 2004).

²⁸ *Ibid.*, [018].

its application to software. In examining open source culture and software arts practice and making explicit reference to Benjamin's 'The Author as Producer' essay, Josephine Berry Slater describes the practice of hiding source code as narrowing the creative potential, and enforcing a series of mythologies around creativity and property rights.²⁹ With a rejection of private property in mind, a good example is 0100101110101101.org's project *life_sharing* (2000-) that rendered the data on a networked computer public property: 'It sets its kernel free and all the functions that concern it, in the same way as a programmer who frees the source code of their software.'³⁰ In this way, the apparatus is laid bare through what they called 'data-nudism'. For Berry Slater, this approach confirms an engagement with code and the relations of production and allows her to question what constitutes a radical work of software art in the context of previous claims for engaging with the technical apparatus. If art holds radical potential at all, the question remains how to produce art that resists its seemingly inevitable commodification and how to reconcile the failure of the avant-garde to deliver its promises. Do the tactics associated with dialectics require complete overhaul or better implementation?

Responding to Eric Kluitenberg's 'Transfiguration of the Avant-Garde/The Negative Dialectics of the Net' (2002) in which he argues for avant-garde tactics to be deployed in the larger context of the network society,³¹ Duna Mavor would consign dialectics to the 'rubble-heap of history' (alluding to Benjamin's historical materialism):

'Dialectics never died. It lives every time another tired exhibit of the relics of dada or situationism opens at the houses of culture across the world. It lives when the hackers who haunt the net repeat the slogans and gestures of the dead and then congratulate themselves when they are finally inducted into the halls of power of the Venice Biennale or Ars Electronica.'³²

She is bitingly cynical about the interventions of art-activist groups such as the 'anti-corporate corporation' RTmark and condemns their strategies as tired repetitions of obsolete logic leading to inevitable recuperation.³³ Through their constitution as a corporation, they sell mutual funds for corporate sabotage and employ a form of limited liability such that there is a certain amount of protection from risk of prosecution. Thus, they effectively use corporatism against itself, by developing a critique from within the system, both using and abusing its powers. Perhaps she is right to be suspicious of RTmark's activism if it invites interest from the commercial art world but also their rejection and auction of their invitations to take part in the prestigious Whitney Biennial is a notable attempt to resist this. Nevertheless, unlike

²⁹ Josephine Berry Slater, 'Bare Code: Net Art and the Free Software Movement', in Geoff Cox & Joasia Krysa, eds., *Engineering Culture*, New York: Autonomedia, 2005 [2002], first published for 'NetArtCommons', *Gallery 9*, Walker Art Center.

³⁰ see http://www.0100101110101101.org/home/life_sharing/index.html

³¹ Eric Kluitenberg, 'Transfiguration of the Avant-Garde/The Negative Dialectics of the Net', quoted in Duna Mavor, 'avant.garde - tranfigured or dead?', *Nettime*, (March 2002). Mavor's position is in keeping with Peter Bürger's notion of the 'post avant-garde' (1984) to describe the failure of the historical avant-garde. Perhaps it is worth noting that she is writing from a post-communist context.

³² Mavor, *Ibid*.

³³ RTmark, <http://www.rtmark.com/>

Mavor's difficulty with dialectics, perhaps the problem with the tactics of RTmark might be that they are not dialectical enough. The need for negation is evident but there is a danger of simply making a joke about politics, in far too straightforward a fashion that can easily be recuperated.

The point is not to repeat the tactics of the avant-garde but to reinvent them. The comparison of software art to earlier avant-garde movements, and particularly the avant-garde activities of the 1920s in Russia and Germany, provides historical understanding of radical forms and strategies. In the contemporary situation, it appears that many of the claims of the avant-garde have become 'embedded in the commands and interface metaphors of computer software. In short, the avant-garde vision became materialized in a computer'.³⁴ On the surface, it seems that what was once a radical aesthetic vision to reveal the social structure behind the visible surfaces has become a standardised form through the use of computer technology. This is in keeping with the contemporary culture's reliance on appropriation, wherein recycling, re-working, and re-combining media are standardised techniques, and confirms the contemporary concern with transformation. So if, as Manovich thinks, software has naturalised montage techniques, how can software be further developed as a radical project in revealing the ideological processes at work? Whereas Manovich emphasises a non-dialectical 'anti-montage' of digital compositing in which elements are blended into a whole rather than brought into collision, Richard Wright argues for the possibility of a digital aesthetic that amplifies the dialectical method.³⁵ His assertion is informed by Benjamin's concept of allegory in which new understandings emerge through the bringing together of historical fragments.

Exploring some of these ideas, Wright's project *The Bank of Time* (2001) produced under the name Futurenatural, is a screensaver that makes allegorical comment on idleness and growth. Wright notes how the germinating plant is a recurring metaphor in financial and investment advertising, as well as in Baroque imagery. The user's idle time is directly proportional to the rate of growth of the plant on their desktop - from seedling to fully grown plant through to its decay. In *The Bank of Time*, the more idle the user the faster the plant grows suggesting idleness at work as a creative force. The pointlessness of all the labouring is made evident. In dialectical allegories such as this, objects are brought together through montage to disrupt the continuity of ideological conceptions. It is a technique to rewrite history according to a different set of principles using montage to force these fragments out of the historical continuum. The principles of montage thus take on a wider political status in the context of a history of art, and perhaps particularly in respect to an artwork that uses software. Is it possible to speculate and apply the principle of software to history in a similar manner by paying attention to its processes and events?

Benjamin's assertion is that it is simply not enough for artists to demonstrate political commitment however revolutionary it may seem, 'without at the same time being able to think through in a really revolutionary way the question of their own work, its

³⁴ Lev Manovich, 'Avant-garde as Software', in Stephen Kovats, ed., *Ostranenie*, (Frankfurt: Campus Verlag 1999), also http://www.manovich.net/TEXTS_04.HTM

³⁵ Richard Wright, *Montage - Transformation - Allegory: A Study of Digital Imaging in Dialectical Film Making* (unpublished PhD thesis, London Guildhall University 1998).

relationship to the means of production and technique'.³⁶ This goes beyond simply an engagement with the apparatus or being satisfied with finished works, but seeking to transform the apparatus because only in this way can the relations of production be transformed too. In the case of software this holds, in that it is not enough to simply reveal source code, make it free or to stress its potential aesthetic form, but it needs to be opened up to further transformation. Self-referentiality is not enough in itself - it must be combined with an improved apparatus. Predating 0100101110101101.org's *life_sharing* and with similar intentions, was the browser *Manifest* by 46liverpoolst.org, released in 1999.³⁷ Using the browser rendered the user's hard drive public in the spirit of a rejection of private property (advocated by Marx and Engels in *The Communist Manifesto* printed at 46 Liverpool Street, London in 1848). In this way, both the software and the hardware were made open source. The User's guide was simply a reprint of *The Communist Manifesto* in the form of screen shots of the text in the browser.

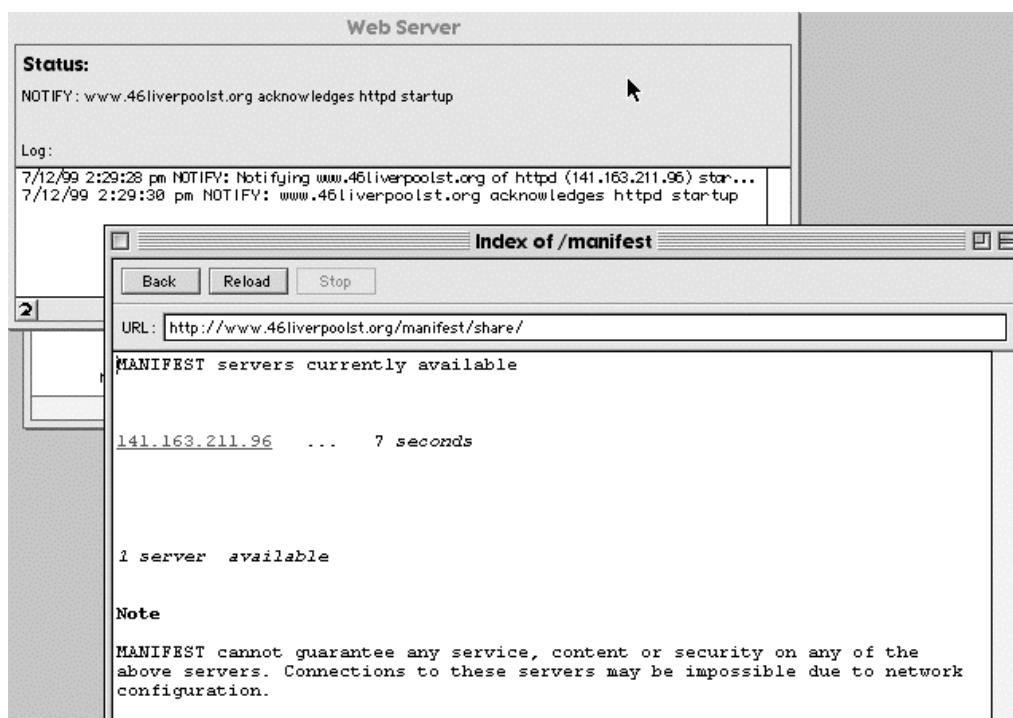


image caption: 46liverpoolst.org's *Manifest* browser (1999)

A dialectical approach combined with an understanding of systems, continues to remain a useful concept and model of change to describe systems that appear to contain the same logic. This approach provides the possibility of change through applied technique. To intervene in the production process in the spirit of Benjamin, the artist-programmer therefore needs to take account of the generative possibilities of the system itself, in recognition of the transformation of the mode of production into something increasingly software-like. Crucial to this method is the retention of contradiction through disorder, for this is where politics is evident and where re-invention takes place.

³⁶ Benjamin, 'The Author as Producer', p. 92.

³⁷ <http://www.46liverpoolst.org> [the project is now offline].

Note:

This essay is a further development of 'Generating orderly disorder: after the author as producer', presented at *Marxism and the Visual Arts Now*, conference, University College London. A related paper also was derived from this: 'Generator: the dialectics of orderly disorder', *Creativity & Cognition*, conference proceedings (ACM Press, 2002), pp. 45-49.

Bibliography

- Benjamin, Walter, 'The Author as Producer', in *Understanding Brecht*, trans. Anna Bostock, (London: Verso, 1992).
- Benjamin, Walter, 'The Work of Art in the Age of Mechanical Reproduction', trans. Harry Zohn, in *Illuminations* (London: Pimlico 1999 [1936]).
- Berry Slater, Josephine, 'Bare Code: Net Art and the Free Software Movement', in Geoff Cox & Joasia Krysa, eds., *Engineering Culture*, New York: Autonomedia, 2005 [2002].
- Bourriaud, Nicolas, *Relational Aesthetics*, trans. Simon Pleasance & Fronza Woods, (Dijon-Quetigny: Les Presses de Réel, 2002).
- Burnham, Jack, 'Systems Esthetics', in *Artforum*, vol. 7 no. 1 (September 1968).
- Guattari, Félix, *Chaosology*, Sylvere Lotringer, ed., (New York: Semiotext(e), 1995)
- Hardt, Michael & Negri, Antonio, *Empire* (Cambridge Mass.: Harvard University Press, 2000).
- Hayles, N. Katherine, 'Chaos as Orderly Disorder: Shifting Ground in Contemporary Literature and Science', *New Literary History*, no. 20 (1989).
- Joxe, Alain, *Empire of Disorder*, (New York: Semiotext(e), 2002).
- Manovich, Lev, 'Avant-garde as Software', in Stephen Kovats, ed., *Ostranenie*, (Frankfurt: Campus Verlag 1999), also http://www.manovich.net/TEXTS_04.HTM
- Mavor, Duna, 'avant.garde - tranfigured or dead?', *Nettime*, (March 2002).
- Owens, Sue, 'Chaos Theory, Marxism and Literary History', in Jody Berland & Sarah Kember, eds., *Technoscience, New Formations*, no.29, Summer (London: Lawrence & Wishart, 1996).
- Virno, Paolo, 'Virtuosity and Revolution: The Political Theory of Exodus', trans. Ed Emory, in Paolo Virno & Michael Hardt, eds., *Radical Thought in Italy*, Minneapolis: University of Minnesota Press, 1996).
- Wark, McKenzie, *A Hacker Manifesto* (Cambridge, Mass.: Harvard University Press, 2004).
- Wright, Richard, *Montage - Transformation - Allegory: A Study of Digital Imaging in Dialectical Film Making* (unpublished PhD thesis, London Guildhall University 1998).
- Wu Jie, *Systems Dialectics* (Beijing: Foreign Languages Press, 1996).

Abstract:

Walter Benjamin in 'The Author as Producer' (written in 1934), recommended that the cultural producer intervene in the production process, in order to transform the apparatus. This essay asks if this general line of argument of thinking retains relevance for cultural production at this point in time – when activities of production, consumption and circulation operate through complex global networks served by information technologies. The dynamic and algorithmic routines of the networked computer are now crucial to the production process in general, and it follows that critical attention should be given to its inner workings and the software on which it relies. An underlying assumption of the essay therefore is that software is not simply a functional tool but expresses wider cultural and technological processes that extend the critical potential of arts practice. The argument is that once radical techniques such as montage need to be rethought in relation to systems theory. Both systems theory and dialectics share a common interest in conceptualising the material world in terms of processes - as the interaction between parts of the system and the development of the whole system that expresses dynamic interactions and contradiction. A critical practice in software art holds the potential to intervene in the production process in the spirit of Benjamin, in recognition of the transformation of the mode of production into something increasingly software-like. Crucial to this is the retention of contradiction through disorder, for this is where politics is evident and where re-invention takes place.

--

Biography:

Geoff Cox is an artist, teacher and projects organiser as well as currently Lecturer in Computing at University of Plymouth, UK. He has a research interest in 'software art' (<http://www.anti-thesis.net/>) expressed in some critical writings and projects, such as the co-curated touring exhibition *Generator* (2002/03). He recently co-edited *Economising Culture* and *Engineering Culture* as part of the DATA browsers series (Autonomea 2004 & 05). He is also a trustee of the UK Museum of Ordure.