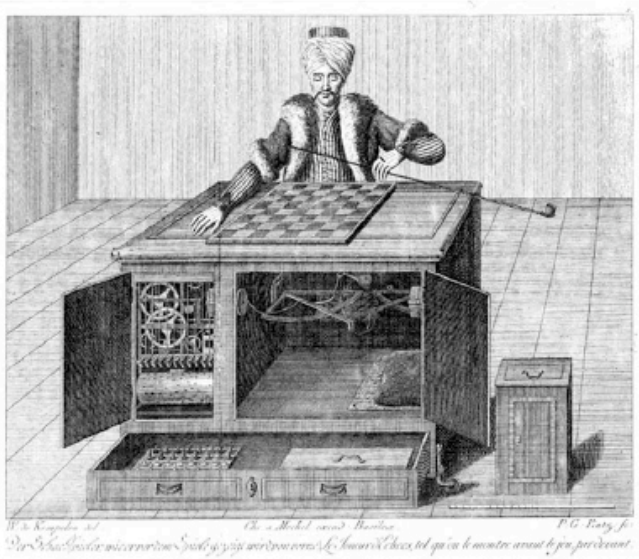
Software Art Has No History

Geoff Cox



'The story is told of an automaton constructed in such a way that it could respond to each move in a game of chess with a countermove that ensured him victory. A puppet in Turkish attire, and with a hookah in his mouth, sat in front of a chessboard placed on a large table. A system of mirrors created the illusion of a table transparent from all sides. Actually a hunchback dwarf, who was an expert chess player, sat inside and guided the puppet's hand by means of strings. One can imagine a philosophical counterpart to this device. The puppet known as "historical materialism" is always supposed to win. It can easily be a match for anyone if it ropes in the services of theology, which today, as the story goes, is small and ugly and must, as it is, keep out of sight.'

[Walter Benjamin, 'Theses on the Philosophy of History', transl. Esther Leslie

In this short paper, I intend to highlight two key issues with relevance to the 're:place' conference and the purpose of historical work in general: firstly, that an analogy can be drawn between the operations of software and historical processes; and secondly (and as a consequence), a term like software art (as a contingent description of practice) should not represent a further art historical genre but more productively offer a means of breaking the continuum that much (media) art historical work seeks to establish.

The title is ironic: borrowing from John Roberts's *Art Has No History!*, in turn based on Althusser's 'Ideology has no History', that itself is a reference to Marx and Engels's *The German Ideology* (of 1845-46).[1] The deliberately playful title belies the obvious fact that art has plenty of history.

When Althusser claimed ideology had no history, he was expressing what he perceived to be its unchanging structure, expressing 'no history of its own'.[2] Roberts's use of the phrase 'Art Has No History' attempted to

---

[1] John Roberts, 'Introduction', to Roberts, ed., *Art Has No History! The Making and Unmaking of Modern Art*, London: Verso (1994: 1-36); Louis Althusser [1969] 'Ideology and Ideological State Apparatuses: Notes Toward an Investigation', in Slavoj Zizek, ed., *Mapping Ideology,* London: Verso, (1997: 100-140); Marx & Engels [1845-46] 'The German Ideology' in Robert C. Tucker, ed., *The Marx-Engels Reader*, New York: Norton (1972: 146-200).

[2] Its sense of history is a mere reflection of 'real history' (in Althusser,

playfully reveal some of the paradoxes of art history: 'that there is no such thing as *art* history, *Art History* and art *history*'.[3] That debate might be developed across and between the three aspects reveals the lasting usefulness of historical materialism (with some revisions) to critique the assumptions of traditional art history for the study of culture, and the dynamic and uncertain interplay between theory and practice.

Much the same applies to the field of software art and culture, where practices resist easy categorisation and historicisation - and the figures of artist, programmer, critic and historian have become fluid. In such a scenario, the artist-programmer is not simply doing work that becomes the object of history but intervening in the very processes of history - as Raymond Williams puts it (in his essay 'The Uses of Cultural Theory'), reflecting 'the socially and historically specifiable agency of [(software) art's] making' (in Roberts 1994: 36).

Although referring to William's 'cultural materialism' in passing, I am evoking the older reference to historical materialism, or what Marx and Engels refer to, in *The German Ideology*, as a 'materialist conception of history' based on the general premise that there are social forces that intervene in the process of history.[4] From this, derives the

---

1997: 122). Louis Althusser actually challenges some of the central tenets of classical Marxism and the centrality of the economic base (that it determines the superstructure) by adding levels of feedback. To Althusser, writing in 1969, the superstructure (that contains culture) is both relatively autonomous and exerts a reciprocal action on the base (1997: 105). This is important as it stresses the politics of culture, and the effectiveness of what he calls the 'ideological State apparatuses' to describe the mechanism of ideology to make things appear natural. Althusser asserts that there is no ideology outside subjectivity, and he includes himself and the reader in this scenario as both thoroughly 'in ideology'. To Althusser, we are 'always-already subjects' practising the rituals of ideological recognition: 'all ideology hails or interpellates concrete individuals as concrete subjects' (1997: 130). It interpellates or recruits subjects by hailing 'Hey, you there!' (1997: 131). The Internet operates in this way too: 'it hails you, it connects to you and gives you an IP number; it interpellates you into Imperial ideology' (Brian Holmes (2003), 'Artistic Autonomy and the communication society', *Nettime*, Oct 26; conference paper for *Diffusion: Collaborative Practice in Contemporary Art*, Tate Modern, London.).

[3] Roberts (1994:1). The first issue (art) derives from a sociological view of art, the second (art history) with what at the time was referred to as 'new art history' to critique the assumptions of art history as a relatively fixed (conservative) field of inquiry, and the third (history) with a more fluid understanding of historical processes.

[4] Humans 'must be in a position to live in order to be able to "make history"'. In other words, the first historical act is the production of the means to satisfy essential needs to live: 'the production of material life itself. And indeed this is an historical act, a fundamental condition of all history...' (1972: 155-156). The emphasis is on the importance of ideas or the active role of individuals in history. In 'The Eighteenth Brumaire of Louis Bonaparte' of 1851-2, Marx claims: 'Men [sic] make their own history but they do not make it just as they please; they do not make it under circumstances chosen by themselves, but under circumstances directly encountered, given and transmitted

dialectics of labour relations - the central paradigm in which history is perceived to be a process between humankind and nature, mind and reality, present and past - and where social antagonism arises. In the present 'social factory' (to describe the way in which the mode of production has been extended to the whole of society) these antagonisms have not disappeared but have been distributed. What was once considered the living contradiction of labour relations in the factory have been extended by collectivity and networked communications technologies to the whole of life. Software lies at the heart of this.

***

Evidently, software art has a history too. It has become fairly commonplace to situate the contemporary term software art in the historical context of the *Radical Software* journal published by the Raindance collective (launched in 1970), and Jack Burnham's exhibition *Software, Information Technology: Its Meaning for Art* at the Jewish Museum, New York (also 1970). In acting upon hardware, software operates as a metaphor for an emphasis on social processes that involve an engagement with relations of production, and how the internal logic of the operating systems of art (and non-art) respond to ***feedback*** from human subjects.[5]

The illusion is that human agency is not involved in these operating systems of art, art history, and history. The task of the historical materialist is to reveal these inner workings in order to develop a counter strategy to received history — to brush it against the grain as Benjamin puts it, in order to 'make the continuum of history explode'.[6]

---

by the past.' (in *Karl Marx and Frederick Engels: Selected Works in One Volume*, London: Lawrence and Wishart, 1980: 96). Hegel, as Žižek points out, would reject the view that human subjects make their own history as far too deterministic. The statement does not take account of the ways in which inner essence can be transformed into external conditions and vice versa.

[5] The *Radical Software* journal 'imagined a world in which the contest of ideas and values could take place freely and openly' outside of the existing institutional and ideological frameworks of commercial telecommunications (David A. Ross, 'Radical Software Redux', http://www.radicalsoftware.org/e/ross.html (2003). In parallel to this, in the *Software* exhibition, Burnham refers to non-object based art and time-based based practices such as performance, interactive and conceptual art, but also public interaction that breaks down the false distinction between the operating systems of art and non-art. This represents the abstract 'internal logic' of a program receiving feedback from human subjects. In this respect, the influences of information theory (associated with Claude Shannon and Warren Weaver), systems theory (associated with Ludwig von Bertalanffy), and cybernetic theory (associated with Norbert Wiener) can be identified.

[6] Walter Benjamin, 'Theses on the Philosophy of History' [written 1940, first published 1950], trans. Harry Zohn, in *Illuminations*, London: Pimlico (1999: 253). He calls this 'Jetztzeit' (the presence of the now): 'History is the object of a

Any moment in time can be traced historically in order to reveal its constructedness, and hence reveal the possibility of change in the present. But also Benjamin argues that any conception of history changes with the times, as does its analysis in accordance with changes in the material mode of production:
'It is the particularity of *technological* forms of production (as opposed to art forms) that their progress and their success are proportionate to the *transparency* of their social content.'[7]

Although Benjamin is referring to glass architecture, the formulation can be readily applied to other technologies to emphasise the availability of either open or closed social content – for instance, as with free software and proprietary models of software production.

The significance of the materialist presentation of history forces the present into a critical state. It is as if time stands still, and the past and the future converge not harmoniously, but *explosively*.[8] Software, like history, expresses predictable and unpredictable tendencies, and emergent potential.[9]

An understanding of emergence that takes into account dialectics is something that 'critical realism', associated with Roy Bhaskar, attempts to achieve.[10] Bhaskar considers society as both the condition and outcome of human agency, and that human agency both reproduces and

---

construction, whose site is not that of homogeneous and empty time, but one filled with now-time' (Esther Leslie's translation of Benjamin).

[7] Walter Benjamin, *Selected Writings: Volume 2, 1927-1934*, trans. Rodney Livingstone et al, Michael W. Jennings, Howard Eiland & Gary Smith, eds. Cambridge, Mass.: Belknap Press (1999: 465).

[8] According to Max Horkheimer too, change can only come about through a 'rupture in the continuum of history' (echoing Benjamin's phrase from 'Theses on the Philosophy of History').

[9] In very general terms, emergence describes a process by which complex patterns are formed from simple rules. This can be a dynamic process occurring over time, but to be considered emergent it should generally be unpredictable. Emergent phenomena can be explained through systems theory, and more particularly through an understanding of adaptive behaviour.

[10] The objectives of critical realism are ambitious (to say the least) in developing a general theory of dialectics that extends beyond Hegelian thinking, to form a critique of Western philosophy. In general, critical realism suggests that the realms of physics and history share a false perspective on natural science, in as much as one tends towards a causal explanation (positivism) and the other an interpretive understanding (hermeneutics). Bhaskar's revisionist position (particularly in his _Dialectic: Pulse of Freedom_ of 1993) is based on the view that Hegelian thinking is closed rather than open-ended, and that Marx never fully described scientific realism. His description of the dialectic in Marx as scientific, explains the contradictions in society in terms of the contradictory relations generating them as historical (rooted in the changes of the circumstances described), critical (demonstrating historical conditions) and systematic (tracing the historical conditions back to the mode of production). Margaret Archer, Roy Bhaskar, Andrew Collier, Tony Lawson, Alan Norrie, eds. *Critical Realism:*

transforms society.[11] Human agents are described as actively able to transform society and yet simultaneously constrained by society (in this way making Marx's position more complex).

Emergence, for Bhaskar, is the generation of new possibilities.[12] Echoing earlier descriptions of historical materialism, the 'here and now' is characterised by the influence of the outside and the past, in such a way that social phenomena can be seen to contain emergent properties (as part of an open system). Emergence in this way describes the creative, autopoietic operation wherein new properties are generated out of pre-existing material forms. Emergence is useful as it suggests non-causal, non-teleological formations and allows Bhaskar to conceptualise human agency in terms of incompleteness and an ongoing transformative (dialectical) process. This is what he refers to as 'transformative agency'.[13] An 'extra-historical materialism' is implied in the way that transformative agency stresses the importance of ideas and the active role of people in historical development - recognising that people do not simply make their own history nor are determined by history, but both.

The important point here, as with contemporary software art practices (or indeed 'social software' non-art practices more generally) is that the human subject is an active part of this. History, and the history of technology, is full of the use of trickery to make it seem beyond the scope of human intervention.

When applied to software, part of the issue lies in better expressing the multiple processes that are running on a computer, and not least the relation between the writing, compiling and running of program code, in order for the programmer and user to position themselves as an active part of the system. In live-coding performances, for instance, the program performs with the programmer, both relaying instructions and acting upon

---

Essential Readings, London: Routledge (1998: xxi).

[11] Bhasker explains: 'Social structure, then, is both the ever-present condition and the continually reproduced outcome of intentional human agency.' (1998: xvi)

[12] Roy Bhaskar, Andrew Collier and Alan Norrie, 'Dialectic and Dialectical Critical Realism' section, in Margaret Archer, Roy Bhaskar, Andrew Collier, Tony Lawson, Alan Norrie, eds. _Critical Realism: Essential Readings_, London: Routledge (1998: 564).

[13] Bhasker (1998: 638). Bhaskar interprets the historical materialist position as teleological (too causal) in its characterisation of progress towards a better society. In contrast, he describes society rather obliquely as an: '... open-systemic entropic totality, in which results [...] are neither autogenetically produced nor even constellationally closed, but the provisional outcome of a heterogeneous multiplicity of changing mechanisms, agencies and circumstances' (1998: 600). An understanding of adaptive complex systems informs this view, and undermines any teleological understanding of history.

them in an uncertain relation.[14] Human agency is foregrounded, and any additional sense of agency assigned to the program relies on the relation of its existence as both expression and process. The act of coding also becomes a prototype for its further action.[15] In other words, coding work - writing, compiling, and running code - comes to represent software as a whole.[16] Is history similarly coded?[17]

Finally, when it comes to software art, it should be remembered that this is not simply to be taken teleologically either: that what is being described is not software to make art but software as art. To historicise software art therefore would undermine its very nature — its inherent dynamic and emergent properties. The suggestion is that software might be similarly conceived of as in a critical state where its past construction and future execution remain in tension. Its source code both expresses what it will do, and can do it at the same time — a movement from in-itself to for-itself in Hegelian terms. Thus, the inner potential and outside influences of an object are continually held in contradiction, between what is possible and what exists. These processes are thereby understood as dynamic and emergent phenomena that are analogous to the inner workings of wider (cybernetic) systems that express ongoing processes of development and feedback.

The approach to history and/both software I have attempted to briefly outline is one that aims to reinforce the argument that software art should not simply be placed within an art historical straightjacket - for instance as a further example of previous work or as a new genre - but that it should be seen as an opportunity to rupture the historical continuum — an opportunity for past and future to converge *explosively*.

---

[14] Moreover, a computer program undermines the distinction between its function as a score and its performance. The program is not detached from its potential performance.

[15] Here I am making reference to Adrian Mackenzie's (2005) 'The Performativity of Code: Software and Cultures of Circulation', http://www.lancs.ac.uk/staff/mackenza/papers.php

[16] This is indeed a definition of software: the program as well as the other material required for the program to run which together is referred to as software. Indeed, software art might draw attention to any one or combination of these activities, but in general is considered to be not the artwork resulting from software, but software as artwork. Software art appears to be well-suited to comment upon the ways in which these processes (associated with art and work) increasingly utilise software but also 'act' like software.

[17] This does appear to be the case with memory. In the technological lexicon, memory applies to: RAM (random-access memory) where programs are created, loaded and run, in temporary storage. Whether these are written to hard memory, into the computer's hard drive becomes a useful analogy to the ways in which working memory is written and more specifically how collective memory is produced. Clearly all sorts of temporarily stored memories are deleted. History is full of such examples (as is Media Art History). From a technical perspective, the computer can be reduced to a

This is something to bear in mind at conferences (such as this) that do historical work.

***

In terms of an introduction to this session 'Cybernetic Histories of Artistic Practices', perhaps what I have described (particularly with reference to emergence) goes some way to imagine what form a *cybernetic history* might take - wherein feedback and agency are essential components.

---

problem of memory: as a machine capable of writing, reading, storing, and deleting data.