Wayne Clements, Chelsea College of Art and Design, London, UK

Art and Software Entropy

Keywords: art practice, computer art, software, entropy, transience, digital preservation

"*But our years will be complete only when they have all moved into the past*."
Augustine, Confessions, Book XI

This paper emerges from my experience as an artist who writes his own software. Why, after a period of time, did my software sometimes seem to go 'wrong'? *Was I to blame*? These questions, I believe, should have relevance not only to artists or software writers, but also to anyone who has had a piece of software fail on them unexpectedly and wondered what it was, if anything, they had done to cause the problem, and what they might do about it.

There are in fact several terms that apply to the tendency of software in general (not just 'software art' or 'net art') to deteriorate and stop working over time. 'Software rot', or 'code rot', or 'software decay' are referred to as forms of 'software entropy'. The multiplication of terms seems to indicate that this may be a well-known phenomenon, even if perhaps less familiar in art and art historical circles. It is the idea of *software entropy* particularly that I will explore, as this concept suggests a rational and scientific basis, and thus one that must be viewed seriously as a problem in art conservation. Also, I wish to explore the question of whether cultural activities, such as computer-based art, should be thought of as breaking – or following – cultural and scientific laws, and what this might imply for art practice and preservation.

– – –

The reasons that a previously functioning program may go wrong are many. Terms such as 'software rot', for instance, do not imply actual decay, but refer to a number of processes whereby software becomes inoperable. This fundamental failure cannot be overcome by such strategies as rebooting the system. The software ceases to function unexpectedly, and either must be 'fixed' or replaced.

Jacobson, Christerson, Jonsson, and Övergaard (1993) attempt to formulate this tendency in terms of the Second Law of Thermodynamics.[1] This law accounts for facts such as why a tyre will go flat if it gets a puncture (and so equalising its air pressure with that of outside), or why a hot plate will cool (through the dissipation of its heat) if left.

> The second law of thermodynamics, in principle, states that a closed system's disorder cannot be reduced, it can only increase or possibly remain unchanged. A measure of this disorder is entropy. This law also seems plausible for software systems and we can assume that this law is plausible for the systems discussed here; a system's disorder, or entropy, always increases. We can call this software entropy.[2]

Jacobson *et al* are careful to advise caution in the application of this comparison. Nevertheless, can the analogy be useful? Does software, and by extension anything with a software basis, have an inherent tendency to deteriorate? 'Anything with a software basis' must include much of contemporary, 'Western' socio-cultural activity.

Artists and art historians are used to the fact that artefacts do deteriorate and require conservation and care. Such attention may not always offset the tendency of artworks to decline physically, and this too may be part of the artist's thinking. For instance, speaking of her Latex sculptures in a 1970 interview with the writer and critic Lucy Lippard, Eva Hesse remarked, 'Life doesn't last, art doesn't last, it doesn't matter'.[3] Lippard notes that, at the original time of writing (1976), 'at least three pieces have disintegrated'.[4] Hesse, of course, is particularly well-known for her conscious adoption of this very fugitive substance. However, is there also something peculiar to *software* that guarantees a catastrophic career towards deterioration and final (and not too distant) destruction? It would follow from this, if so, that artists who use or make software are expressly embracing the transient.

To answer this, we must evaluate, to make the point again, the force of Jacobson *et al*'s analogy: does software suffer entropy and does this inevitably effect a destruction of any software-based, or software using, artwork?

Before I respond to this problem, I would like to explain its background, at least for me. The reason for raising these issues lies in a conversation with a curator. I have referred to this conversation elsewhere but without identifying the curator by name.[5] I wish to acknowledge him here because I am indebted to him for obliging a consideration of what for me has proved to be a productive line of investigation. He is Perttu Rastas, a curator of new media at the Museum of Contemporary Art, Finnish National Gallery (KIASMA). This is a summary of our exchange of views:

> At a recent conference there was discussion of the preservation of new media art. One of the panellists, an esteemed curator of new media at an important national museum in Scandinavia, spoke about conservation and the need to preserve old computers and programs and the difficulties of achieving this. Nevertheless, he believed the fundamental project was viable. Someone proposed that an obstacle to preservation was the artist's carelessness. This view was shared by several of the panellists. I replied from the floor that much Internet artwork uses events and content from remote websites over which the artist and the artwork have no control. I said that this artwork was inherently unstable and temporary. It was intended to be so, and this was part of its unique quality. It cannot be preserved. The museum curator replied that I had decided (I think I quote accurately), 'to choose to break cultural laws'.[6]

Despite having referred to the discussion before, I do not want to appear to take what were, after all, a few remarks made in debate out of context and so perhaps blow them out of proportion. The reason I continue to raise the matter is that I find suggestive the implication that to follow cultural laws is to break, or at least to resist, physical laws. The opposite seems equally suggestive: to break cultural laws is in some sense to follow physical laws. I am thinking of the Second Law of Thermodynamics and its effect of undoing the existence of order — in this case, cultural order based on the preservation of physical structures.

Jacobson *et al* refer specifically to a *closed* system's disorder. But are artworks, and computer artworks particularly, closed systems analogous to closed physical systems? This, it should be asserted at the outset, must be doubted, particularly if we refer (as above) to software and artwork that 'uses events and content from remote websites over which the artist and the artwork have no control'; and also if we consider networked machines and software more generally. This is to make a distinction between 'standalone' computers (which are not connected to other machines) and networked machines. A networked machine (and networked art) should in a sense be thought of more as an *open system.* We must now consider whether the Second Law of Thermodynamics can apply to such a system and, if so, in what ways.

Software may be 'open' in many different ways. It may, for instance, receive inputs from remote computers and users in the ways mentioned in the quotation above. One risk here is that these interfaces may accelerate entropic processes. Not only will the software experience its own entropy, as described by Jacobson *et al*, but it may also receive contributions from the other computers with which it interacts. (It should be noted that physical systems are not usually thought of as absolutely closed or open, rather these should be considered tendencies.)

It is therefore not clear that more or less open systems are more likely to resist entropy than are less open systems. Whether it increases or decreases depends on how much entropy is shipped out, compared with how much entropy is imported.[7] The contrary is also true: the order in a system cannot increase faster than it is imported. In practice this depends on many factors. I will expand upon this shortly.

I wish briefly to turn to the idea of cultural laws. Do these laws exist, and can – and should – we break them? I take it that these laws are *not* literal, such as those that protect intellectual property; or R.A. No. 1265, an Act which makes the flag ceremony compulsory in all educational institutions in the Philippines; or the various laws relating to Listed Buildings in the UK. These laws exist and they may be cultural laws, but they are, I think, not the ones in question here. I presume that what is being discussed in fact is the attempt by human societies to construct and preserve forms of cultural order (in the sense of relatively stable structures), which survive over duration of time and extent of space. It is this, expressed as 'law', that is violated in the embrace of the openly impermanent and, probably, short-lived.

Of course there are art forms based on performance where this transience is known and accepted (much of dance, musical performance, the theatre, and so on). It cannot be these that are the matter here. Context surely must be our clue: the fine arts, in so far as we refer to the visual arts, have traditionally a durability that we have come to expect and rely on, at least to some degree. In this sense Perttu Rastas has a case that is worth considering. Art preservation is clearly in a long term battle with the forces that will undo both physical and cultural structures. In this, he could be thought to be correct: culture works against the Second Law of Thermodynamics. Angrist and Hepler (1967), for example, write: ‘In a certain sense the development of civilization may appear contradictory to the second law’.[8] They go on to argue that the success of these attempts at resistance is qualified at best: ‘Even though society can effect local reductions in entropy, the general and universal trend of entropy increase easily swamps the anomalous but important efforts of civilized man’.[9] Angrist and Hepler argue, ‘Each localized, man-made or machine-made entropy decrease is accompanied by a greater increase in entropy of the surroundings, thereby maintaining the required increase in total entropy’.[10] Even if we do not accept these arguments, the point is made that cultural order works against processes that effect to undo it.

Considering these cultural norms, set against the issue of entropy, I wish to address what might be done about entropic effects in computer art. What can we do about software preservation and the artworks that depend upon it? I have noted already the difference between open and closed systems. One possible problem for our arguments is the identification of cultural systems with *closed systems*, where entropy is likely to increase and cannot decrease. Yet is it feasible that *open systems* (in contrast to closed) might import order, and thus export entropy more quickly than they import it? Can the chaotic effects of the increased entropy of closed systems be thus offset?

In practice, artworks may often be thought of as physically closed systems. The traditional task of art preservation may be considered that of reducing the openness of the artwork’s physical system to a minimum. The aim of preservation is therefore to reduce the effects of the physical interaction of the artwork with its environment to the smallest amount. How viable is this strategy with computer-based art? These artworks exist in the technological environment, and this is in a process of continual and rapid change. Furthermore, this process of change is widespread and includes operating systems, specific applications, and web-based resources. Changes, and consequent lack of accord between any one of these, can lead to an overall failure of function. (Computer-based artwork shares with other applications this technical requirement to function fully.) Perhaps the most obvious problem, concerning operating systems, is lack of backward compatibility. Such a problem has affected Microsoft’s software in the past: what will work on an old version of the system may not work on a new. Specific applications, however, may also not keep up with the changing environment around them. For example, this ‘alert’ is what I get when I attempt to start a very well-known instance of Internet art – *Web Stalker* (Fig. 1).

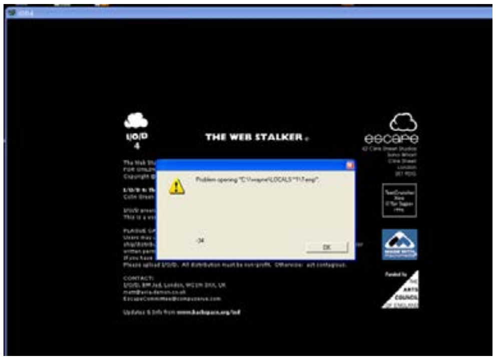
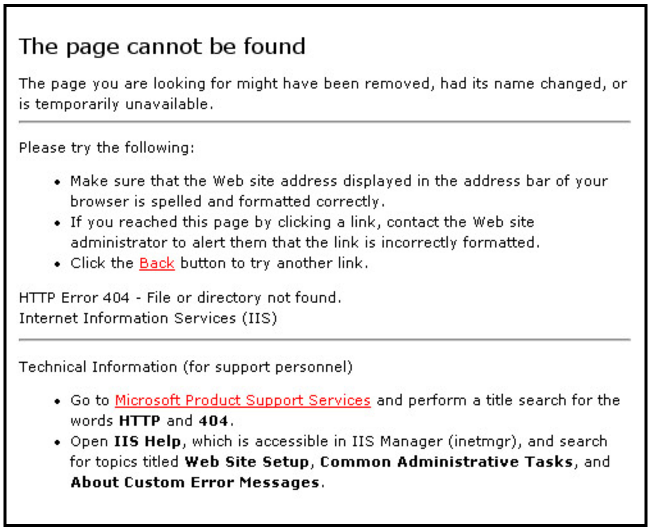


Fig. 1. Web Stalker (screenshot)

This is because the application had not been optimised for Windows XP, which I was using at the time of writing.

There are several of my own works which have not run properly because the material that they once used has been moved or has disappeared from the web. There are often ways of accommodating these changes. Sometimes this is not possible and we are likely to receive some sort of error notice instead (Fig. 2).



The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- Make sure that the Web site address displayed in the address bar of your browser is spelled and formatted correctly.
- If you reached this page by clicking a link, contact the Web site administrator to alert them that the link is incorrectly formatted.
- Click the Back button to try another link.

HTTP Error 404 - File or directory not found.
Internet Information Services (IIS)

Technical Information (for support personnel)

- Go to Microsoft Product Support Services and perform a title search for the words **HTTP** and **404**.
- Open **IIS Help**, which is accessible in IIS Manager (inetmgr), and search for topics titled **Web Site Setup**, **Common Administrative Tasks**, and **About Custom Error Messages**.

Fig 2. 404 Error.

That these are facts about the life of art on the Internet can hardly be doubted, but what might be done about it? Bearing in mind our consideration of the remorseless battle between chaos and order above, any proposal must be cautious in the extreme.

I wish to conclude this paper with a qualified endorsement of open source agreements as a promising – although by no means certain – response to the issues outlined above. It is possible to publish source code, of course, without the legalities of open source agreements. But it is likely that these may be adopted if the intention to distribute is serious and further upkeep and development by others is desired. In any case, publication has, I wish to contend, the potential, at least with art that is software, to extend the longevity of the artwork. Publishing source code will not help the ‘hardware’ aspect of art, in all probability. The outcome, it must be repeated, is not assured: publication will not compensate for certain of the disruptive effects of the importation of entropy identified above such as the removal or absence of web resources. There is also a chance that the more hands involved with maintenance and upkeep, the more risk there is of the importation of unforeseen effects, of well-meaning but unwanted changes.[11] Openness here may be at the possible price of the importation of entropic effects which could lead to the hastened deterioration of software structures.

There are several sorts of open source agreements. I will not try to evaluate or describe their relative merits. The one I use is the GNU General Public License. It reads:

> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.[12]

There is the possibility, at least, that if the details of a program are published, that this may allow the reversioning of the software by other interested parties. It should here be said again: this is a possibility, not a certainty. Much depends on the long-term reception of the publication. It depends on *interest*.

Code alone may not be essential: even if the code is lost, if a sufficient description should exist, it may allow a reconstruction of the artwork, such as I demonstrated at CHArt 2004 with the recreation of *COMPUTERIZED HAIKU*, a work from 1968.[13] I cannot go into this more deeply here, but the argument holds: the structure of the work, if known, may be preserved against forces that could destroy it. This is more likely if the artwork's construction is knowingly shared.

The idea of artworks as capable of distribution by means of publication of source code is relatively new. It is not yet mainstream, but may become so as it is adopted by artists who exhibit in mainstream art galleries. One such is Cory Arcangel, whose work *Colors* is exhibited at the time of writing (2009) at Tate Liverpool's *Colour Chart* exhibition. Its code may be found at http://www.beigerecords.com/cory/Things_I_Made/Colors. It may be used by others to make their own video works. We must wait to see if this model will be more widely adopted.

In this paper the idea of cultural forms of order has been set against the effects of entropic processes that may impinge on software. It was argued furthermore that the more closed a software system is, the more likely it is that its entropy will increase, or certainly not decrease. The possibility of 'exporting' entropy was entertained as one means of maintaining computer-based artwork through the exploitation of open source development.

It is possible that the adoption of an open approach to the creation and preservation artworks is a good model for the future of art that comprises, or uses, software to a considerable extent. This may be best regularised by the use of open source agreements.

It is by now apparent that this approach does not assure us of success in off-setting the effects of entropic deterioration of our software-based cultural pursuits, whether these are artworks or something else. Much depends on the interest and goodwill of others to maintain the software. As noted above, even with goodwill, this too may have unwanted effects. To echo Hesse, perhaps in the long run nothing lasts. But it is up to us to decide how much it matters and how we should respond.

November 2009

References

[1] Jacobson, I. Christerson, M., Jonsson, P., and Övergaard, G. (1993), *Object-oriented software engineering: a use case driven approach*. Wokingham: Addison-Wesley.
[2] Jacobson, I. Christerson, M., Jonsson, P., and Övergaard, G. (1993), *Object-oriented software engineering: a use case driven approach*. Wokingham: Addison-Wesley, p. 69.
[3] Lippard, L. (1992), *Eva Hesse*, New York: Da Capo, p. 210.
[4] Lippard, L. (1992), *Eva Hesse*, New York: Da Capo, p. 210.
[5] Clements, W. (2008), ‘Surveillance and The Art of Software Maintenance: Remarks on logo_wiki’, *Observatori 2008. After the Future*, Valencia: Observatori.com.
[6] Clements, W. (2008), ‘Surveillance and The Art of Software Maintenance: Remarks on logo_wiki’, *Observatori 2008. After the Future*, Valencia: Observatori.com, p. 42.
[7] Sewell, G. (2005), ‘Appendix D. Can “ANYTHING” Happen in an Open System?’, *The numerical solution of ordinary and partial differential equations*. Hoboken: Wiley, pp. 316-320.
[8] Angrist, S. W. and Hepler, L. G. (1967), *Order and chaos: laws of energy and entropy*. New York: Basic Books, p. 210.
[9] Angrist, S. W. and Hepler, L. G. (1967), *Order and chaos: laws of energy and entropy*. New York: Basic Books, p. 210.
[10] Angrist, S. W. and Hepler, L. G. (1967), *Order and chaos: laws of energy and entropy*. New York: Basic Books, p. 210.
[11] Jacobson *et al* (1993, p. 71) states of their analogy, that the ‘main difficulty with this kind of “law” is, of course, that people are involved’.
[12] GNU General Public License, http://www.gnu.org/copyleft/gpl.html (accessed 8 September 2009).
[13] Clements, W. (2004), ‘Computer Poetry’s Neglected Debut’, *Futures Past: Twenty Years of Arts Computing*, Proceedings of the 20th Annual Conference of Computers and the History of Art (CHArt) held at Birkbeck College, University of London, 11–12 November 2004, A. Bentkowska, T. Cashen and H. Gardiner (eds), http://www.chart.ac.uk/chart2004/papers/clements.html (8 September 2009).