DIGITAL

OBJE

CT

S

MATTHEW FULLER

Digitality and objects

If software has a social and technical imaginary, if it is culturally active as a force in itself, what does that mean for data more generally, the objects constructed, giving rise to, or handled by software?

In a recent interview Michel Serres[1] suggests that a typical contemporary development is a drive experienced in science to aim towards an understanding of the specificity of an object. Earthquake-causing tectonic faultlines or individual livers are rendered, by various methods and by the peculiar capacity for differentiation typical of digital analysis, as something with individual qualities and traits rather than generalised or diagrammatic instantiations of a 'type'. Every scanned liver, every library book in a database, every phone, person or every mapped asteroid is also a digital object.

Under digitalization more generally, there is a widespread tendency for all objects, processes and qualities to become transduced by data-gathering, patterning and identification. Take a virus, this little darling is known by virtue of an electron microscope to be so cute and cuddly and roundishly polygonal with just enough weird fuzzy bits and clotted fraying edges to make it amenable to love. It looks like another planet. One you could escape to, off earth, find death. And this is the thing about an object, no matter how much of a specificity it can be recognized as being, it is not alone. David Wojnarowicz writes, "When I was told that I'd contracted this virus it didn't take me long to realize that I'd contracted a diseased society as well."[2]

Over a decade after his death in New York on the 22ND August 1992 I want to hold on to this quote from Wojnarowicz, in fact all of his work, because it stands as a reminder of much of what is missing from debates around software, a thick, brilliant, absolutely enraged, vividly sexual and gregarious involvement with multiform life. Software is part of this, but not much. That there are programs such as iLife or MyLifeBits that would have you believe so, life suffused with the toxic long-chain polymer aroma of fresh white computer plastic, there are programs such as SAP which would like to have you not notice their structuration of your life. The X-rays from hell that Wojnarowicz writes about are not simply that of the perfectly named and unique human organ, they are about the tenderness and corporeality of living things, bodies. When, at birth or before, you contract the diseased society that you live in, you, as an object, are in connection with a million relations of dimensionality, permutational fields of being.

But what Wojnarowicz's text also shows is that things don't just all fit. There is no necessary total inter-relation of all parts with every part a mirror and a node of the universe with a pre-established harmony between all substances. There are, as a shorthand, enormous, insensible gaps between knowable parts. This essay does not attempt to remedy such a situation. What is hoped is to set out some of the terms in which objects are composed in recent software projects in which the nature of the digital object is recognized as being significant.

It has been established that software has a politics, an aesthetics, that these elements and compositions of algorithmic logic invent, deploy and make stable different kinds of sociability and inter-relation with other elements. However, such general concepts need to made more supple and detailed. If materials have their own capacities, which may or may not match the scalar models which we have of them and by which we arrange behaviour with and through them, what does this mean for digital objects? How are they separate, at what scale, and how can these permutational fields, their dimensions of relationality be sensed, made palpable and used?

Temporality

The work of Walter Benjamin, performing "a sort of spectrum analysis"[3] on the complex relations of history and futurity embedded in and coursing through the fruits of the arcades, is perhaps significant here. Objects carry utopias with them, imaginary worlds of ideal use and misuse, of combination with other elements in a planned or tasteful grammar, of imaginary relations to the world as modern or bucolic, or as easy, sophisticated, charming, portable, educational, ornate and so on. In Benjamin the sacred and predictable path in time from the lower to the higher is apparent in some texts, as in those moments when he is more fiercely Marxist. In others, as in the text where he most clearly states his conception of historical materialism,[4] this philosophy appears as a weave, with threads weft in and out of the warp, to disappear, reappear, changed in their context. It is in some sense an immensely horizontal, parallel vision of time, time as a carpet, as well as one that includes a notion of work and of process, a woolly cellular automaton, seething with bugs, stray threads and knots 'as an afterlife of that which has been understood and whose pulse can be felt in the present'.[5] But it is also one, 'directed towards a consciousness of the present which explodes the continuum of history'.[6] That is, it is an approach to time that is thick, that senses the muffled thunder and still present sparks in objects and ideas 'of the past' and that draws the question of what is to be done into the present with a sonorous and stinging sense of the possible.

We might ask, what is temporality to a digital object? We know that versioning forms one sort of clock-setting in both proprietary and free software that is interwoven with that of hardware manufacturer, but it is also clear that some software goes explicitly sideways in relation to such orderly time (think of Dyne:bolic, a Linux distribution written only for low-end, cheap, Pentium processors.[7]) Asking how software produced using art methodologies operates in relation to time is not simply so that we can confirm the existence of dull retro-styled work, but also to see it as a potential dimension for escape and invention, for tinkering. As standard software marches onwards with its own feature-panicked elegance it draws a line which in part, describes the space of what is possible and which at the same time allows the impossible or the improper to find itself more easily. The development of these projects and others allows the terrain of the digital imaginary to stretch and reflect.

Composition

Shared with Benjamin's understanding that the historical object constitutes a problem in itself is the understanding that an object is never in itself complete. One software project I worked on as a collaboration with Graham Harwood, *TextFm*[8] showed that technical organizations call for social and medial organizations around them, certain kinds of work and activity in order for them to occur. This work couldn't simply be aimed as a downloadable at an individual user on the other side of a PC, but needed organizations, people to work on it, to put up transmitters, use radios and phones, use their knowledge of a city to make news spread in the right way and so on.

But that sense of assemblage doesn't simply stop at the casing of a computer, or at the head and foot of a script. Elements which are commonly understood to be simply technical—that is to have the horizon of their dimensions of relationality being as described in the accompanying standard object documentation—are here understood as also having capacities drawn out of them according to context and composition. All three of the projects discussed here also have something of this quality, they address more than just a single user, but imagine or invent a social process.

Aside from social organisation that each object induces and is embedded in we can also mask this question on a much smaller 'internalist' scale: what relations are developed between software art and what say ICT disciplines might call 'content', text, image files, and so on, but also, what are the ways the software makes elements available for use, hides things, functions or processes as internal to itself, or treats this as that which is passing through, which it work on, that give it purpose, but which are not themselves software? So organisation passes from one scalar state to another, but as it moves, does not stay the same. At each scale, and there are many, different relations of dimensionality come into composition. Social software that reflects upon and incorporates visions of the social that are repressed in mainstream software is a particularly interesting place to begin to look at such issues.

Nine

Nine is a web based application developed by Harwood for Mongrel.[9] Based on the web, and useable via a browser, it works as a lightweight multimedia collaging structure.

Nine builds on the fact that handling algorithmic materials and logical and procedural processes have become part of peoples' general skills. That is, that people may not necessarily have skills in using computers directly, but they are used to handling social processes that have been reformulated for the benefit of easy computerization. In some contexts this is not necessarily a benign skill, we have all seen parts of life ripped open by computerized arithmetico-material drives founded upon quantification and calculability. But, as a social software[10] project, nine allies such skills with the capacities people have to, as Arjan Appadurai says, live in several social imaginaries simultaneously.

On opening the Nine website you are faced with a grid containing patches of colour. Each install of nine sits on a server and contains a pre-coded maximum number of uses. The square of squares that you see first when logging-in sets this, a very visible constraint, twenty-seven squares on each side of a square. Each smaller square representing a 'map' of nine images. This set of nines continues, with each image having nine potential hot spots to which data such as sound, text, video, internal link or a zoom can be added. One of the things that becomes apparent very fast in the use of nine is that the structure of permissions and of hierarchies that sits underneath most software is very much at the surface here, whilst the software is aimed at being 'as light as possible'[11] it is extremely rigid about the roles of users and of data. Using the software or reading the nine help file,[12] you easily become aware of how this software sets each data element in a recursive procedure of sets, permissions and process. The repetitive nature of the work is clear, it is there to force through a thinking about structure. The constraint to nine images is a way of cutting through digital abundance. It means that some kind of selection must be made. When it comes to the nine hot-spots though, it seems as if this may already be too many. I've not yet seen any nine maps with all of these 'used up'. Is the rigid arbitriness of the number nine as a governing principle tight enough? Nine by nine as a set of commitments to produce material is pretty substantial. At the same time, there are users clearly building up significant archives of material, such as the images, sound and text on Congolese music by Vince.

The means of differentiation of objects are quite clear. How the project then stitches its parts together is rather interesting. Whilst there is no facility to add credits to images, other than in a text link, in order for one Nine 'knowledge map' to embed an image from another, it is necessary for the owner of that map to email the person who is responsible for the one in which the picture sits. The system sets up a simple need for people to network, but more importantly to register that their images have a longer-term life than a quick afternoon's work. Users have often been intrigued by the meanings ascribed by others to their images, the links that are made of them. In another form of link, the system keeps a flat list of every word used. Map maintainers are prompted by email whenever another user uses the same word. Semi-automated links between discrete texts can thus also be set up. There is thus an interesting interplay between different layers of the database its ongoing use and the possibility for registering common terms or ideational nodes across time and place.

The three software projects here each develop a particular set of mechanisms for collaborative work on data. All three are also in whole or in part written for and arise from workshops in which someone with previous experience of the software, often the producers, works with others who may often have less experience in direct use of computers. Although the softwares are written for a potentially 'universal' usage, they are often particularly local in their application. Nine for instance is often used, and used well, by Imagine IC[13] a small institution concerned with experiences of migration sited only a few minutes walk from where the software was written in Amsterdam's Bijlmer suburb. Crucially, these projects do not formulate themselves in terms of a question of 'access' or as a response to the too rigid concept of 'the digital divide', but as a move to generating the means for multiple digital imaginaries to thrive.

Opus commons[14]

Opus Commons is a system developed by Sarai media centre[15] in Delhi and often used for exhibition purposes by Raqs media collective. In a sense the project is an attempt at formulating a concurrent versioning system for digital objects. CVS is a means of storing, tracking, and annotating versions of software or parts of software in a way which allows people to use view and develop code, to produce multiple variations from the same source, and crucially, to go back to earlier versions if errors are made. Such a link to CVS is clear in Opus—the archive of its own code is directly linked to from the front page of the Opus site.

Where the interface approach of nine is to simplify what can be done in order to encourage fast use and to provide extensive documentation for detailed questions, Opus presents the phenomenological difficulty of interface by placing up front an immense quantity of data about each object, its qualities and processing. To those used to the standard behaviour of a GUI removing or graying out redundant functions according to mode, Opus is a significant confusion. It feels like an attempt to carry over the sensibility of the command line to a graphically rich system and so the screen reads as cluttered. The use of graphic background elements and other more decorative elements—compare the linked CVS for code—suggests that the project might benefit from a clarification of its design. At the same time, the presentation of 'more than necessary' information about the nature of the objects it works with is an explicit part of the aesthetic of this system. Within the area of software art or in digital art generally there is a tendency to what might be called ostentatious desublimation, a ponderous revelation of this or that technical truth of a system which replaces metaphysics with the awe of the, often studiously irrelevant, detail. Opus answers this tendency by turning to account, by *making necessary* a consideration of the working culture of digital objects. It is however a working culture that is part actual, part imaginary. Perhaps the dream-user of Opus does not yet exist or if they do, it is quite possible that they have a sophisticated enough imagination of software or media not to need this particular armature.[16]

Opus puts in place a means by which what Serres points to, that is the immense individuation of parts that is inherent to the digital, can be combined with a subtle and compelling means for recognizing and working on variation of use over time. Much of how the system achieves this is through the way in which it arranges its understanding and handling of objects. There is an extensive manual for the project, but the stage is set for the work by a license based on the Gnu Public License.[17] This version of the GPL is integrated into the project at a functional as well as conceptual level: the repertoire of work activities that Opus is concerned with are the capacity to view, download, transform and upload digital objects, with the condition that any such action contributes to an ongoing shared pool of resources.[18]

There is a particular vocabulary[19] used in Opus which marks out its particular attention to both the mechanics and imaginary of uses and changes to a digital object:

Source; Recensions; Project; Themes; Keywords; visualisation.[20]

The first two are concerned with how the particular object sites in relation to its specificity, there is a hierarchy in time beginning with the first manifestation of an object in the system. Themes and projects are to do with the clustering of objects and the clustering of clusters or parts of clusters. Keywords and the schema for visualising the work provide means for the objects to be linked. The concept of a rescension is key to understanding the pattern of work:

"A Rescension is either a re-arrangement of an existing text, or a re-working of an existing text, incorporating new materials, and/or deleting some old ones, or , a new edition with a substantive commentary or annotation.

A Rescension is neither a clone, nor an authorised or pirated copy nor an improved or deteriorated version, of a pre-existing text, just as a child is neither a clone, nor an authorised or pirated copy, nor an improved or deteriorated version of its parents."[21]

Here Opus can be seen as experimenting with another set of dynamics of circulation and linkage of digital objects. Whereas Nine uses a spatially-gridded model that interlinks structures of ownership and permission in order to make collaboration something that has to occur explicitly between users, Opus uses sets of spiraling data. Collaboration occurs through the pool of resources and through the availability of particular kinds of objects rather than through direct contact. One can quibble with the possible consequences of using a biological simile of the parent and child but what is interesting is its significance in terms of an attempt to produce a way of modeling a system on an understanding of digital objects as something that is always in circulation. The claim that information wants to be free is by now familiar, and in many cases, actually useful. What we know now is that this is not a *fait accompli*. The inherent capacity of digital objects to be circulated needs a little help, conceptually as well as in terms of technology. It is perhaps in its metaphorical inverting of the usual computer science understanding of the parent and child as describing respective standing in rank (in a directory or network topology for instance) that the significance of this project can be sensed in relation to time. Time becomes resonant and more aware of the fields of permutation and possibility that course through it and make it. The achievement of Opus is in finding a way to begin such work, but more important is the means by which it does so. That is, it is not simply theoretically mellifluous, but tests and composes itself also at the level of practice.

Every element has a short description and full description of their 'meaning' and they are also extensively annotated in terms of size, file type and other details. Because the project is grounded on this sense of circulation, reversioning and weaving of digital objects it pays attention to the cultural implications of file formats, in ways which are often hidden as irrelevant by systems such as those for content management. The project works because it is sensitive to digital objects, I would argue that in part this is a little overdone, resulting in a complicated interface, but at a more fundamental level, which is also more simple, the project works. Key to this is its treatment and understanding of digital objects and the way they are embedded in and engage different working cultures.

Spring Alpha[22]

Serres' brief comment describes a moment in which scientific procedures allow greater insight into material formations, to the understanding that- by virtue of their passage through times, through growth in a particular body, through their contraction of a society[23] — all livers are not of one kidney. It also shows that arithmetico-material drives continue to expand and modulate compositions and elements in the continuing metamorphosis of being by knowledge and in reciprocal or asymmetric oscillations between the two. When it comes to digital objects as elements in software we can, at times, plan to be rather more shallow. Everything is already a standard object. There is no variation between an 'order of nature' and an anomaly which either escapes from that order or contributes to marking out the space of the known and possible.

How can social software projects establish the conditions for other forms of knowledge to become manifest and active in the use of the digital objects they make available? To put it another way, what happens to the special understanding the poppy farmer, the food manufacturer and the architect have about the objects they take part in generating when you find an object that links them: say, a few wraps of heroin hidden from moisture in a crisp packet and from sight between loose bricks in a wall at the back of a pub?

How is it possible to make a systematic (for in software there can be nothing else) domain in which such variations in the use, meaning, capacities and conjunction of objects becomes possible? Alternately, how can the thickness of meaning, of pasts, of repressed potencies which Benjamin senses in objects in relation to time also be made palpable in the seemingly ever-on presentness of digital objects?

Spring_alpha is an open source gaming system currently in the early stages of development, and a project which in part attempts to deal with some of these problems. How is it possible to make a shared model of an imaginary social conflict, a revolutionary situation, when the everyday objects which actor-network theory shows as providing a hidden factor of stability in social relations are even more fundamentally reified by virtue of their being the super-standardised models and associated behaviours afforded by a digital system?

How flexible is a digital object? You cannot reform a nuclear power station, the inherent danger, and social requirements of such a system mean that they should be dismantled. Such a technology provides one extreme, but what about an orange polythene traffic cone? Can you throw it, wear it as a hat, shout through it? Given this, how can you generate a computer model of an object and those it might come in contact with supple enough to afford such uses? To put it another way, how can you generate a software architecture supple enough that the programmer or game designer does not have to imagine all possible uses or scenarios, but allows them to emerge through the interactions of inventive users? (This is not the same as a form / content question such as that in which an email client 'contains' all possible emails.) How can objects be dumb enough to be complex?

Given that some form of simplification, reduction and exaggeration is necessary or at least inevitable, how can the startling lives of objects when the world is turned upside down, their own propensities for turning and being turned be made palpable? The game, it should be remembered, is not a simulation, and in its current prototype phase Spring_Alpha has an intriguing suggestion. Composed in a way that is explicitly modular, the landscape and artifacts of the housing estate are to have their properties visible and manipulable, to have their properties and capacities emerge out of some mechanism of use. Every object, such as a traffic cone, factory, house, cop or plate of food will also include something along the lines of a patch familiar from audio and video manipulation programs such as Pure Data[24] or Isadora.[25] Patching systems provide a sophisticated way of representing functions, procedures and relations between elements. Each object is a box potentiated with functions and processes. The user creates a flow chart of boxes which function to generate simple, or vastly complex interactions. Simon Yuill who is leading the development of this project expects much of the specific qualities of the program to emerge from workshops in which characters, street objects and buildings are assigned qualities to be remodeled as such 'patches'. Participants will work through the material culture of their surroundings and imagine their reinvention. The project is in the very early stages of development so the first version of this software will be the real point at which it can be substantially discussed, but as a model this promises to find one way of combining the synthetic powers of software with those of the social in a particularly compelling way.

This short, relatively naïve, account of the characteristics
and interconnections of digital objects in these projects
finds sustained and detailed work being done not simply
on software itself, but on what it handles, what it is
for and by which it is co-produced. Further
work on the terms and dynamics of the
differentiation and conjunction of
digital objects would take such an
account into realms of intense
details but also find a way
through the unbearable
massiveness of what
has been con-
tracted
.

Thanks to:

Graham Harwood, Matsuko Yokokoji, Mervin Jarman and Mandie Beuzeval for discussions about nine; Shuddabrata Sengupta and Lawrence Liang for high-speed info on OPUS Commons; and Simon Yuill, Francis McKee, Eleonora Oreggia and Chad McCail for information on early stages of Spring_Alpha.

1 See, Michel Serres and Peter Hallward, 'The Science of Relations: an interview', in, Angelaki, vol.8. no. 2 issue editor: Peter Hallward, Routledge, Oxford, 2003, p.231

2 David Wojnarowicz, 'Postcards from America, Xrays from Hell', in, *Close to the Knives, a memoir of disintegration*, Serpent's Tail, London 1991, p.114

3 Walter Benjamin, 'Paralipomena to 'On the Concept of History'', in, *Walter Benjamin, selected writings, volume 4 1938-1940*, trans. Edmund Jephcott et al., eds., Howard Eiland and Michael W. Jennings, Harvard University Press, 2003, p.402

4 Walter Benjamin, 'Eduard Fuchs, Collector and Historian', in, *Walter Benjamin, selected writings, volume 3 1935–1938*, trans. Edmund Jephcott, Howard Eiland et al., Eds., Howard Eiland and Michael W. Jennings, Harvard University Press, 2002, p.260-302

5 Walter Benjamin, 'Eduard Fuchs, Collector and Historian', p.262

6 Walter Benjamin, 'Eduard Fuchs, Collector and Historian', p.262

7 see http://www.dyne.org/

8 http://www.scotoma.org/TextFm/

9 The software is written in Perl under the GPL, the current working version was finished in 2002 and is currently in preparation for a possible new round of development.

10 Social software as a term is used here in the sense of the 2002 essay, *Behind the Blip*. However, it is useful to recognize the subsequent use of the term for the more narrowly defined sense of software

which is used for social networking and analysis. A useful overview of this area, amongst others is maintained by SebPaquet http://www2.iro.umontreal.ca/~paquetse/cgi-bin/om.cgi?Social_Software

[11] Harwood, in conversation, May 2004

[12] *http://9.waag.org/Help/* Nine is extremely clearly documented, primarily through minimally annotated images.

[13] http://www.imagineic.nl/

[14] Opus Commons' main site is at http://www.opuscommons.net/ This site is currently being updated and is thus only partially functional. A use of the system developed for the show *How Latitudes Become For ms: Art in a Global Age* curated by Steve Dietz is available at http://opus.walkerart.org/

[15] http://www.sarai.org/

[16] One feature of Opus is that there is a set of requirements for someone to register as a user. Possibly, the design of the user as an object is perhaps too present as a schematic to be inviting of easy initial use. As with nine, how the user function is entangled with the author position—and the repressions and opportunities this affords—by this registration process would be worth considering.

[17] See, for the text of this license: http://www.gnu.org/copyleft/gpl.html

[18] For a discussion of open content licenses in general see, Lawrence Liang's forthcoming review of the spectrum of such licenses at http://pzwart.wdka.hro.nl/

[19] A related project is Raqs Media Collective, *A Concise Lexicon of/for the Digital Commons*, available at http://www.sarai.net/compositions/texts/works/lexicon.htm

[20] More detailed descriptions of these terms are given in the Opus Commons user manual

[21] from Opus Commons manual

[22] http://www.spring-alpha.org/

[23] Natalie Jeremijenko's 'One Tree' project is interesting in relation to this. Several hundred clones of one tree are sited at sites with a range of different socio-ecological features. Mapping what happens to these no longer identical trees over the years of their lives forms the core of the project.

[24] http://pure-data.sourceforge.net/

[25] Isadora software is at *www.troikatronix.com*
Thanks to Scott de la Hunta for a demo of this work.