All Problems of Notation Will be Solved by the Masses:
Free Open Form Performance, Free/Libre Open Source Software, and Distributive Practice

Simon Yuill

In recent years, the foregrounding of 'collaboration' in artistic practice has acquired an aura of inherent benevolence and emancipation, as though the very act of working with others in itself ensures some form of resistance, or alternative, to conventions of cultural production. The recent valorisation of collaboration within the arts, however, merely elides the basic condition of collaboration that all forms of production ultimately rely on in various degrees and arrangements. This can be seen as one part of the larger growth in service and communications industries whose 'labour' and 'produce' are primarily invested in the structuring and intensification of various collaborative exchanges, often minute and ephemeral, yet, when harvested on a vast scale, capable of generating seemingly endless amounts of profit.[1] Collaboration in this form of production extends beyond the contracted employees into the consumers themselves, who help define and create the products they themselves consume. This is exemplified in the proliferation of highly 'personalised' products and services, reality entertainment, and the social networks of Web 2.0, with the virtual world of SecondLife notably combining all three factors.[2] Those artforms which most consciously foreground collaboration, as described in Bourriaud's Relational Aesthetics, merely echo this situation.[3] The social relations constructed by the artist in gestures of collaboration with audiences and others become spectacularised and commodified in forms that often do not return to those who created them but rather become tokens circulated within the art

[1]
The relation of profit to surplus value in regard to so-called 'immaterial labour' have been much contested in recent debate, particularly in light of the controversial rejection by Negri of the application of Marx's 'law of value' to modern economies. For a critical overview of these issues, and a response to Negri, see: George Caffentzis, 'Immeasurable Value? An Essay on Marx's Legacy', The Commoner, issue 10, Spring/Summer 2005, http://www.commoner.org.uk/10caffentzis.pdf

[2]
SecondLife http://secondlife.com

[3]
Nicolas Bourriaud, Relational Aesthetics, translated by Simon Pleasance and Fronza Woods, Paris: les Presses du Reel, 2002.

[4]
For one perspective on this see: Saul Albert, Who Will Be Transformed?, http://twenteenthcentury.com/saul/who_will_be_transformed.htm

[5]
See: Eleonora Belfiore, 'Auditing Culture: the subsidized cultural sector in the New Public Management',

market.[4] In a funding system that prioritises social inclusion within the arts, like that of the UK, collaborative projects can tick the box that unlocks the piggy-bank of state patronage. In such contexts, collaboration quickly becomes little more than a revenue stream.[5] Similarly, the rise of relational aesthetics has accompanied the embrace of artistic practice by the commercial sector, often drawing upon the strategies of such art to enhance collaboration and 'creativity' within the workplace.[6]

For some, Free/Libre Open Source Software (FLOSS) appears to offer a model of practitioner-led collaborative practice that, through its legislative mechanisms such as copyleft licensing, could be applied to artistic practice in a way that might counteract such problems of recuperation.[7] An initial enthusiasm for this, however, has given way to disenchantment as the application of FLOSS to artistic practice appears to create more problems than it solves.[8] These problems have arisen through an emphasis upon issues of collaboration and legislation that often fails to recognise the proper relation of these to FLOSS's primary mode of production – the notational medium of code.[9] Enthusiasts for a FLOSS inflected approach to artistic practice have similarly failed to properly consider forms of cultural practice that have been emerging from within FLOSS and how these may relate to other forms of cultural production outside of that community. A consideration of these reveals that such practices are not so much collaborative but rather distributive. Rather than accumulating and cohering the labour of others they enable capacity for self-production elsewhere. Through a comparison of current FLOSS-related arts practices to related earlier artistic forms this article outlines the relation between notational production and distributive practice.

Livecoding

Of all the artforms supported and enabled through FLOSS, 'livecoding' has emerged as the one which most directly embodies the key principles of FLOSS production in the creation and experience of the work itself. In livecoding the artwork is expressed in software code that is written and re-written live during its performance. Many livecoding artists write their own software tools to support this way of working. Alex McLean's 'feedback.pl' was one of the first such tools.[10] It is a simple Perl script that continuously reads and executes an extract of its

International Journal of Cultural Policy, 10.2, 2004.

[6]
This was a model promoted by think-tanks such as DEMOS in London in the late 1990s, the artist Carey Young both works within and exposes this area.

[7]
One of the earliest such statements was Saul Albert, Open Source and Collective Art Practice, http://twenteenthcentury.com/saul/os.htm

[8]
These issues were brought up in the discussions looking back on the NODE.London festival, see: http://www.mazine.ws/NodeL_NMC, Al Altendrof, 'Calling Time Out', http://altendorf.enemy.org/en/works/texts/calling-time-out.html, and the 'NODE.London Evaluation Report', http://eval.nodel.org

[9]
An initial example of such an approach can be found in Felix Stalder, 'On the Differences between Open Source and Open Culture', Media Mutandis: a NODE.London Reader, edited by Marina Vishmidt, Mary Anne Francis, Jo Walsh and Lewis Sykes, London: OpenMute and NODE.London, 2006.

[10]
Alex McLean, 'Hacking Perl in Nightclubs', http://www.toplap.org/index.php/Hacking_perl_in_nightclubs, originally published in 2004 at http://www.perl.com/pub/a/2004/08/31/livecode.html

[11]
For an overview of livecoding issues see the TOPLAP website: http://www.toplap.org, as well as discussions on the TOPLAP and OpenLab (London) mailing lists: http://www.pawfal.org/openlab

[12]
fluxus: http://www.pawfal.org/fluxus, PacketForth: http://packets.goto10.org

[13]
SuperCollider:

own code displayed in a text editor. This code defines various algorithms from which music is generated. During performance this is re-written by the performer, changing the musical structure and effectively improvising from within the code. A projection of the performer's desktop makes this visible, thereby emphasising how the code and the changes made to it are integral to the work and to the audience experience of it. The material and formal relationships between code and music are therefore discernible, even though many audience members may be unfamiliar with programming languages themselves.[11] To some extent this is comparable to witnessing a performance on an acoustic instrument such as guitar or clarinet. Whilst we may not understand how to play such instruments ourselves, we can relate the gestures of the performer to the sounds that we hear and thus acquire a sense of the relation between the sound and its material production. This contrasts sharply with previous forms of electronic music performance, such as those of Jean Michel Jarre and Todd Machover, in which interface devices are presented on stage often simulating and referring to acoustic instruments. Livecoding dispenses with such 'fetishes' and is unashamed to expose the bare materiality of its production. The unfamiliarity of presenting code as a raw material, however, results in something very different from that of the guitar or clarinet performance, and more akin to revealing the stage machinery in a Brecht play. It creates a virtue by exposing something that is normally concealed.

Whilst livecoding has initially developed as a form of music, it is not restricted to this. David Griffith's 'fluxus' and Tom Schouten's 'PacketForth' are tools for creating visual works, the first based on a 3D graphics engine and the second a video processing system.[12] Some existing tools, such as SuperCollider, Chuck and Pure Data have also been used for livecoding work.[13] In fact, any programming language or tool that can execute code on the fly can potentially be used for livecoding. The concept has also been extended into other forms of work. 'Social Versioning System' (SVS) enables multiplayer simulation games to be created and coded live, with new code distributed amongst the players as a game evolves.[14] Ap's 'Life Coding' is a large scale performance combining software coding, circuit bending and conference-style spoken presentations.[15]

Livecoding Aesthetics

http://www.audiosynth.com, Chuck: http://chuck.cs.princeton.edu, Pure Data: http://www.puredata.info

[14] http://www.spring-alpha.org/svs

[15] http://1010.co.uk/xxxxx_at_piksel2007.html

[16] Livecoding differs, for example, from Manovich's model of 'new media' defined primarily as the relation of an artist-created interface providing an aestheticised visualisation with limited forms of manipulation to a database of media such as video clips or statistical info. Whilst presenting an experience that is dynamic, such new media works nevertheless often remain closed unmutable artefacts rather than exposed working processes, see: Lev Manovich, The Language of New Media, Cambridge, MA: MIT Press, 2001.

[17] Piksel: http://www.bek.no, MAKEART: http://makeart.goto10.org, Dorkbot: http://www.dorkbot.org,OpenLab (London): http://www.pawfal.org/openlab, OpenLab (Glasgow): http://www.openlab-glasgow.org, pure:dyne: http://puredyne.goto10.org, FLOSS Manuals: http://wwwflossmanuals.net. The pure:dyne project is a version of the GNU/Linux operating system specifically geared towards artistic usage and supports many livecoding tools.

There are two key aspects of livecoding that embody FLOSS principles. Firstly, the way it makes the continual re-writing of code a primary mode of artistic production and, secondly, in its presentation of the 'work' itself as an open-ended mutable piece of code rather than as a static discrete artefact. In distinction to most non-digital and new media art that is presented solely as a commodity to be consumed, livecoding makes its own materials and practices of production available to others.[16] Livecoding emphasises the FLOSS principle of code-based production as a form of production that is itself 'live' and living, that enables the possibility of production by others for their own purposes.

This 'enabling the possibility of production by others' is often continued outside of performance not only in the use of FLOSS-style distribution, but also in the conscious use of workshops as a means of presenting works and teaching the skills used in their creation. This pedagogic aspect extends to the importance given to technical meetings and development workshops in artist-run festivals such as Piksel and MAKEART, or groups such as Dorkbot and OpenLab, and into the creation of dissemination platforms and projects such as pure:dyne and FLOSS Manuals.[17] The often ad-hoc workshop nature of many livecoding performances and projects themselves is an extension of the livecoding ethic of sharing and making materials generally available. In the case of the ap events that are deliberately staged over long durations of 12 hours or more, this includes participants learning and adapting the tools of the performance as they take place. On a smaller scale, the London OpenLab group host 'drumming circle' performances in which anyone can join in with their own algorithms and code, constructing and developing a collective rhythmic work, as well as performances that start from one piece of code that is rewritten by successive performers. Rather than something marginal or extraneous to the 'art', the idea of the workshop has been absorbed as an integral aspect of livecoding aesthetics.

Livecoding is not the sole or even dominant form of practice pursued by all those involved in FLOSS-related arts. What all practitioners involved in these projects do share, however, is a commitment to the broader notion of 'live code' as a mode of production and a common preference for a workshop aesthetic. It is also within these more 'pedagogic' practices that artistic production within FLOSS meets with other aspects of the FLOSS

[18]
Important early hacklabs include LOA in Milan, http://www.autistici.org/loa, and ASCII in Amsterdam, http://scii.nl, see also: http://www.metamute.org/en/Real-Meet-the-Hackers-Hackers-this-is-the-Real

[19]
The Transhack meetings, for example, began in Italy but have spread elsewhere, the Chaos Computer Club meetings in Germany provide a similar context.

[20]
dyne:bolic project: http://dynebolic.org

[21]
RampArts: http://rampart.co.nr,Hackitectura: http://www.hackitectura.net, Riereta: http://riereta.net

[22]
For information on the Autolabs in Sao Paulo, Brazil, see: Ricardo Rosas, 'The Revenge of Lowtech: Autolabs, Telecentros and Tactical Media in Sao Paulo', Sarai Reader 04: Crisis/Media, edited by Shuddhabrata Sengupta, Monica Narula, et al. Delhi: Sarai Media Lab, 2004, available online: http://www.sarai.net/publications/readers/04-crisis-media

[23]
It might be assumed the 'distributive practice' deliberately echoes or relates to notions of 'distributive justice' but the similarity in the names is purely coincidental. There are many different models of distributive justice, with the liberal model of John Rawls and right-Libertarian model of Robert Nozick being amongst the most widely discussed. Both of these are based around an allocation of material resources.

world, and specifically the political and socially engaged practices emerging from hacklabs and hackmeets.

Hacklabs & Hackmeets

Hacklabs are voluntary-run spaces providing free public access to computers and internet. They generally make use of reclaimed and recycled machines running GNU/Linux, and alongside providing computer access, most hacklabs run workshops in a range of topics from basic computer use and installing GNU/Linux software, to programming, electronics, and independent (or pirate) radio broadcast. The first hacklabs developed in Europe, often coming out of the traditions of squatted social centres and community media labs. In Italy they have been connected with the autonomist social centres, and in Spain, Germany, and the Netherlands with anarchist squatting movements.[18] Hackmeets are temporary gatherings of hackers and activists in which skills, tools and knowledge are exchanged and projects developed. Amongst the first hackmeets were those in Italy in the 1990s.[19] There are direct connections between many of these and artists working with FLOSS. The dyne:bolic project (from which pure:dyne evolved) partly developed through the Italian hackmeets and Dutch hacklabs.[20] RampArts hacklab in London has provided a meeting point for the local OpenLab group, and in Barcelona, spaces such as Hackitectura and Riereta have supported several FLOSS-based art and political projects.[21] Not all artists working with FLOSS and livecoding necessarily share the politics of the hacklabs scene, nor do all hacklab participants necessarily look upon their own activities as art-related, and some are, sometimes rightly, sceptical of artistic involvement in what they do. Hacklabs, however, have been absolutely fundamental to the development of FLOSS in recent years, especially in Europe and South America, and have provided a clear political and ethical orientation in contrast to the somewhat confused and often contradictory political and social perspectives articulated in the other communities and contexts of the wider FLOSS world.[22]

If livecoding is one of the most emblematic artistic manifestations of FLOSS, hacklabs have become one of its most emblematic social forms. Whilst the two may not occupy identical trajectories, they nevertheless overlap and compliment one another in many significant ways. Central to this is their shared principle of 'enabling the possibility of production by others'. This is an issue of distribution, not simply

Where the notion of distributive practice discussed here may relate to distributive justice is in those which are based on, or identify a form of distributive justice in Marx, where the emphasis is upon creating capacity for production amongst others.

[24]
Ornette Coleman, 'Something To Think About', Free Spirits: Annals of the Insurgent Imagination, volume 1, San Francisco: City Lights, 1982, p.117, for other aspects of the free jazz scene see: Bill Cole, 'Improvisation in Music - A Black's View', Free Spirits: Annals of the Insurgent Imagination, volume 1, San Francisco: City Lights, 1982; Frank Kofsky, Black Music, White Business: Illuminating the History and Political Economy of Jazz, New York: Pathfinder, 1998; Graham Lock, Forces in Motion: Anthony Braxton and the Meta-reality of Creative Music, London: Quartet Books, 1988; Benjamin Looker, Point From Which Creation Begins: the Black Artists' Group of St. Louis, St. Louis: Missouri Historical Society Press, 2004.

[25]
English translations have been published in Umberto Eco, The Role of the Reader: Explorations in the Semiotics of Texts, Indiana: Indiana University Press, 1979, and in Audio Culture: Readings in Modern Music, edited by Christopher Cox and Daniel Warner, London: Continuum, 2004. The references in this article are taken from the latter.

[26]
Information about the Scratch Orchestra is taken from the following sources: Cornelius Cardew (editor), Scratch Music, Cambridge, MA: MIT Press, 1974; Cornelius Cardew and

distribution at the level of product, in the way a piece of software can be easily distributed for example, but at the level of practice. The practice itself is inherently distributive, for it integrates the distribution of the knowledge of how to produce into that which it produces.[23] Whilst this allows for possibilities of collaborative production, it should be seen as distinct from collaboration in itself. For whereas a practice that is collaborative coheres the production of many under a single goal, thereby directing the disposition of their labour, a practice that is distributive enables the disposition of labour by others under their own direction. This is facilitated in the output of production as notation, as code that not only creates a product, but enters into an active life beyond its initial implementation.

Notational Production

Notational production is not unique to software. The emergence of livecoding as an initially musical activity reflects the engagement with notational production that has characterised many different musical traditions. The application of computer code to the construction of sound is, in one sense, simply one more episode in this process. Livecoding works from within a particular relation between notation and contingency. The specificity of code is opened towards the indeterminism of improvisation. In this respect livecoding not only adds to the evolution of notational production within music but echoes a particular period where a similar relation between notation and contingency came to the fore. This was a period in which the 'free playing' of experimental jazz developed by the likes of John Coltrane, Ornette Coleman and Sun Ra, met with the 'open' compositional systems of the avant-garde that had been developed by John Cage, Karlheinz Stockhausen, and Earle Brown. Just as FLOSS brings together two related, yet differing, ethics of software production ('Free Software' and 'Open Source'), we might describe this music as Free Open Form Performance (abbreviated as FOFP). 'Free playing' was a term preferred by Coleman and other jazz musicians who rejected the use of the term 'improvisation' on the grounds it was often applied to black music by white audiences to emphasise some innate intuitive musicality that denied the heritage of skills and formal traditions that the black musician drew upon.[24] 'Open' comes from Umberto Eco's 'Poetics of the Open Work', an essay from 1959 which was amongst the first to survey and analyse the experiments with aleatoric, indeterminate and partially

Rod Eley, Stockhausen Serves Imperialism, London: Latimer new Dimensions, 1974, available online: http://www.ubu.com/historical/cardew/cardew_stockhausen.pdf; Stefan Szczelkun, 'The Scratch Orchestra', http://www.stefan-szczelkun.org.uk/phd102.htm; Stefan Szczelkun, '25 years from Scratch', http://www.stefan-szczelkun.org.uk/PHD-SCRATCH2.htm; John Tilbury, 'Cornelius Cardew', in JEMS: An online Journal of Experimental Music Studies, originally published in Contact no, 26, Spring, 1983, pp.4-12, http://www.users.waitrose.com/~chobbs/tilburycardew.html; Michael Chant, 'A Turning Point in Music History', in Experimental Music Catalogue, http://www.users.waitrose.com/~chobbs/chant.html; Michael Chant, et al., Twenty Five Years From Scratch, catalogue for event at ICA London,: London: London Musicians Collective, 1994, available: http//metamute.org/files/25years_scratch.pdf

[27] The constitution is reprinted in Cornelius Cardew (editor), Scratch Music, Cambridge, MA: MIT Press, 1974, pp.10 - 11.

composed works that were emerging in the classical avant-garde.[25] By the late 1960s these two strands of development had crossed over, with jazz composers such as Coleman and Anthony Braxton consciously working with the instrumentation and structural forms of the classical avant-garde, and groups such as the Scratch Orchestra adopting the collective structure of ensembles such as the Art Ensemble of Chicago. Experiments with notation were significant to many of these groups and composers, but in the Scratch Orchestra, the exploration of notational production was a cornerstone of the project.

Scratch History

The Scratch Orchestra grew out of a series of public classes in experimental music that Cornelius Cardew and other composers had been running in London in the late 1960s. These began at the Anti-University on Rivington Street and then at Morley College, a workers education centre set up in the 19th Century.[26] It was here that the original members of the Scratch Orchestra first came together: Cornelius Cardew, Michael Parson, Howard Skempton and people attending their classes. The foundation of the Orchestra was officially announced in June 1969 through the publication in the Musical Times of 'A Scratch Orchestra: draft constitution' written by Cardew.[27] The constitution defines the Orchestra as

> […] a large number of enthusiasts pooling their resources (not primarily material resources) and assembling for action (music-making, performance, edification).

Membership was open to anyone, regardless of musical ability. Many visual artists, such as Stefan Szczelkun, joined and brought with them an interest and experience of art happenings and urban intervention works.[28] Through these, and more conventional concerts, the Orchestra aimed to 'function in the public sphere' presenting works developed by the group. The constitution outlined various forms of activity that the Orchestra would follow in creating these works. One of the most important activities was the writing of 'Scratch Music'. Each member of the Orchestra had a notebook, or 'Scratchbook', in which they would write small works that could be combined

[28]
Stefan Szczelkun was involved in a number of different forms of art collective from the late 1960s onwards, see: Stefan Szczelkun, Exploding Cinema 1992 - 1999, culture and democracy, PhD Thesis, Royal College of Art, 2002, available online: http://www.stefan-szczelkun.org.uk

[29]
Examples of these are reproduced in Scratch Music, 1974, and in Twenty Five Years From Scratch, 1994.

[30]
This does not mean that Orchestra members wrote a new score everyday, however, as the rate of composition would vary between members and over time. The process described here is summarised from the various revisions of the Scratch Orchestra constitution and from information offered by Stefan Szczelkun in personal correspondence.

[31]
As Michael Chant stated: 'no Scratch Music is copyright', Scratch Music, p.17, see also discussion of copyright in Cardew and Eley, Stockhausen Serves Imperialism, p.5.

[32]
Scratch Orchestra, Nature Study Notes: Improvisation Rites 1969, edited by Cornelius Cardew, London: Scratch Orchestra, 1969.

[33]
The Woody Guthrie and Situationist anti-copyright notices are discussed in Dmytri Kleiner, WOS4: The Creative Anti-Commons and the Poverty of Networks, http://info.interactivist.net/article.pl?sid=06/09/16/2053224

[34]
Nature Study Notes, score PDIR3, p.3

[35]
Nature Study Notes, score FRMEVR4, p.3

[36]
The performance is described in more detail at: http://1010.co.uk/xxxxx_at_piksel2007.html

into larger ensemble pieces.[29] The constitution emphasises that these Scratch Music pieces should be an active process of experimentation with different notational forms: 'verbal, graphic, musical, collage, etc.'. By 1972 a clearly defined process for the development of Scratch Music had emerged. Each piece was originally performed by its author, the scores were then exchanged and performed by other Orchestra members, providing a kind of 'peer review' critique of the pieces. 'Scratchers' were asked to write no more than one new piece per day, but encouraged to keep a 'regular turnover', so that there was a tight feedback loop between writing and performing.[30]

From the very beginning the Scratch Orchestra took a conscious decision to make all their notations freely distributable, stating that the Scratch Music works were without copyright.[31] One of their first collections of scores, published in 1969 and called Nature Study Notes: Improvisation Rites, replaced the conventional copyright notice with the following:

> No rights are reserved in this book of rites. They may be reproduced and performed freely. Anyone wishing to send contributions for a second set should address them to the editor: C.Cardew, 112 Elm Grove Road, London SW13.[32]

Whilst rejections of copyright restriction were nothing new, both the Situationists and the folk singer Woody Guthrie had placed anti-copyright notices on their works, it is notable that the Scratch Orchestra also encouraged others to modify and add to their scores, stating that these may be incorporated into the next version.[33]

The works in Nature Study Notes are all textual instruction pieces. Few of them describe ways of making sound however, and instead focus around various social interactions that construct and play with power relations amongst the performers. Some are like party games:

> Form a standing circle. Nominate a leader, who stands in the circle with eyes blindfolded. The remainder of group rotate slowly around him/her. … When the leader is touched, he forfeits his role and

[37] Edsger W. Dijkstra, 'My recollections of operating system design', handwritten memoir, 2000 - 2001, pp.13 - 14, available as an electronic document, EWD1303, from the Dijkstra archives: http://www.cs.utexas.edu/users/EWD/transcriptions/EWD13xx/EWD1303.html, and http://www.cs.utexas.edu/users/EWD/ewd13xx/EWD1303.PDF. See also: Simon Yuill, 'Interrupt', in Software Studies: A Lexicon, edited by Matthew Fuller, Cambridge MA: MIT Press, 2008.

[38] Louis Pouzin, 'The Origin of the Shell', http://www.multicians.org/shell.html, 2000, Paul Graham, 'The Roots of Lisp', http://www.paulgraham.com/rootsoflisp.html, 2002.

[39] Linus Torvalds, 'Notes for linux release 0.01', http://www.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.01, 1991.

[40] John F. Szwed, Space is the Place: The Lives and Times of Sun Ra, Edinburgh: Mojo Books, 2000, p.114.

[41] Jacques Attali, Noise: the Political Economy of Music, translated by Brian Massumi, Minneapolis: University of Minnesota Press, 1985.

[42] For information on origins and development of the LOGO Labs see: Seymour Papert, Mindstorms: Children, Computers, and Powerful Ideas, Brighton: The Harvester Press, 1980, and Cynthia Solomon, Logo, Papert and Constructionist Learning, http://logothings.wikispaces.com, c2007.

[43] Cornelius Cardew, 'Treatise: Working Notes', in Cornelius Cardew, Treatise Handbook, London: Editions Peters, 1970, p.iii.

[44] Cornelius Cardew, Schooltime

so doing shouts 'Porridge'.[34]

Others like generative automata:

> Each person entering the performance space receives a number in order. Anyone can give an order (imperatively obeyed) to a higher number, and must obey orders given him by a lower number. No. 1 receives his orders from the current highest number (the most recently entered player); the highest number can give orders only to No. 1.[35]

Noise Interrupts

Many of the scores in Nature Study Notes set up small scale 'operating systems', simple organisational structures that enable other works to be produced within them. The notion of the performance as an operating system is one that ap have taken up in their Life Coding project. Adapting mechanisms from computer systems, the interaction of performers is dictated by interrupt signals connected to actions defined in look-up tables.[36] In conventional computers, the interrupt mechanism enables signals from peripheral devices such as mice, keyboards or network cards to enter into the operating system. When an interrupt signal is received, the computer selects a response action by matching an identifier code for each signal against a look-up table of programmed routines known as 'interrupt handlers'. In this way pressing keys on a keyboard or moving the mouse can change the course of events currently in action. The interrupt creates a vector between the internal operation of the central processing unit (CPU), the domain of notational operations, and the contingency of the outside world. As Edsger Dijkstra, one of the inventors of the interrupt system, noted:

> It was a great invention, but also a Box of Pandora. Because the exact moments of the interrupts were unpredictable and outside our control, the interrupt mechanism turned the computer into a nondeterministic machine with a non-reproducible behaviour, and could we control such a beast?[37]

Compositions, self-published, London, 1968.

[45]
This is referred to in the Wikipedia entry on Papert: http://en.wikipedia.org/wiki/Papert

[46]
Ivan Illich, Deschooling Society, new edition, London: Marion Boyars, 1995.

[47]
Papert, 1980, op. cit., p.34.

[48]
Ibid., p.5.

[49]
Cardew made this statement as part of his response to the International Symposium on the Problematic of Today's Musical Notation, held in Rome, 1972, see: Cornelius Cardew 'Self Criticism: Repudiation of Earlier Works', in Cardew and Eley, 1974, op. cit., p.88.

[50]
The classic account of hacking and the MIT AI Lab is Steven Levy, Hackers: Heroes of the Computer Revolution, updated edition, London:Penguin, 2001. It is worth noting that Papert was not really one of the 'core' hackers, however the parallels between the practices of hacking and the LOGO Labs are clear.

[51]
Michael Beeler, William R. Gosper, and Rich Schroeppel, 'HAKMEM', Memo 239, Artificial Intelligence Laboratory, Cambridge, Mass: Massachusetts Institute of Technology, 1972. HAKMEM was originally published as an internal memo within MIT AI Lab, a copy of the document is available online: http://www.inwap.com/pdp10/hbaker/hakmem/hakmem.html, and at: ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-239.pdf. Many of the HAKMEM examples also engage with code in a way that conscious exposes the materiality of the machines on which they run, see: Simon Yuill, 'Code Art Brutalism', in READ_ME 0.4, edited by Olga Goriunova and Alexei Shulgin, Aarhus: Aarhus University, 2004.

[52]

The interrupt breaks the closed linear unfolding of the Turing Machine, enabling programs to be stopped, altered and restarted. This enabled the development of languages that could be executed as individual statements one step at a time, giving rise to shell commands (the basic text-based commands used in the UNIX terminal) and the read-evaluate-print-loop (sometimes 'read-eval-print-loop' or REPL for short) that forms the basis of interactive programming languages such as Lisp.[38] The interrupt and read-eval-print-loop lie at the heart of any livecoding program and all UNIX-derived operating systems. In his notes for the first release of Linux, Linus Torvalds wrote: 'interrupts aren't hidden' - a statement that is as much aesthetic as it is technical.[39] It is here where contingency and notation meet, but it is here also that the possibility of error enters. For some, however, rather than treading lightly for fear of a crash, the error carried on an interrupt signal is a positive, productive opportunity. This is not restricted to computer interrupts. During rehearsals, Sun Ra would deliberately interrupt and trick his performers. The 'errors' this produced, however, were not mistakes but rather forms of evolution:

> There are no mistakes. If someone's playing off-key or it sounds bad, the rest of us will do the same. And then it will sound right.[40]

The operating system of Ra's Arkestra incorporated such 'noise' and restructured itself in the process. This 'noise' is not simply that of unmusical sound, but also in the sense that Jacques Attali adapts from information and systems theory, any material that is not recognised by an existing system, and is therefore opposed to 'information' which is material that has value or significance in a given system.[41] Attali describes the evolution of musical styles as one in which an existing system of music becomes exposed to 'noise' that at first disrupts it, but then, through incorporation restructures it and gives rise to a new system. In the voyage of the Arkestra, systems would collapse and be reborn on a daily basis.

Schooltime Compositions

This power over systems was not limited to the Demiurge or intergalactic jazz master. During the same period in which the Scratch Orchestra were re-inventing music from the ground up,

For example:

ITEM 146: MUNCHING SQUARES
Another simple display program. It is thought that this was discovered by Jackson Wright on the RLE PDP-1 circa 1962.

```
DATAI 2
ADDB 1,2
ROTC 2,-22
XOR 1,2
JRST .-4
```

2=X, 3=Y. Try things like 1001002 in data switches. This also does interesting things with operations other than XOR, and rotations other than -22. (Try IOR; AND; TSC; FADR; FDV(!); ROT -14, -9, -20, .)

[53]
Theodor Adorno, 'On the fetish character in music and the regression of listening', in The Culture Industry: Selected Essays on Mass Cuture, edited by J.M. Bernstein, London: Routledge, 1991, p.48. For some, Adorno's teatment of jazz music has been seen as problematic due to the somewhat limited forms of jazz music available through the European music market at that time, if anything Adorno's criticisms exemplify the gulf between jazz as practicised within the black community and jazz as experienced by consumers of the commodified forms that Adorno had access to, for a discussion of this see: Frederic Jameson, Late Marxism: Adorno, or, The Persistence of the Dialectic, London: Verso, 1990, p. 141.

[54]
Cornelius Cardew, 'Towards an Ethics of Improvisation' in Cardew, 1970, op. cit., p.xvii. Nick Collins has outlined a series of training exercises for livecoding: Nick Collins, 'Live Coding Practice', paper presented at New Interface for Musical Expression 2007, New York, available online at: http://www.cogs.susx.ac.uk/users/nc81/research/livecoding practice.pdf

[55]
Conflicts and branches are two common features of version control systems, such as CVS,

a group of children at Muzzey Junior High School in the US were experimenting with their own improvised notation systems. These children were not writing music however, but teaching themselves to program computers. They were part of the first LOGO Lab, a project initiated by Seymour Papert, a researcher from the MIT Artificial Intelligence Laboratory.[42] LOGO was a simple programming language that directed an entity called a 'turtle'. The turtle could either be an on-screen virtual character or a small robot that was instructed to move around their terrain (screen or floorspace) and that could draw a trail on its path. LOGO Lab students developed their own programs in which the turtles would act out drawings or spatial exercises. In so far as LOGO expresses a series of potential actions out of which a drawing emerges it has an analogy to the notations of the Scratch Orchestra, which often did not express sound directly but rather actions from which sound could arise. As Cardew wrote in his notes to Treatise: 'Notation is a way of making people move.'[43]

Like the Scratch Orchestra, the LOGO Labs grew out of a conscious pedagogical interest directed towards developing forms of collective, self-directed practical research. These were realised through semi-structured 'improvisational' activities and used self-developed notational systems as a means of constructing, communicating and reflecting upon these. As the constitution makes clear, the Scratch Orchestra was a conscious exploration of what notation could be and how that related to establishing another understanding of what the practice of music itself might be. This came out of the pedagogic context of the Morley College classes, and, in a perhaps self-mocking gesture, the Orchestra's Nature Study Notes and Cardew's earlier Schooltime Compositions scores deliberately took the form of school exercise books.[44] Papert believed that programming was a skill that should be available to everyone not as a 'technology' – a mechanism for manufacture abstracted from human labour – but as a means of conceptual exploration. There are political parallels between the two projects. Papert had come to computing from a prior involvement in radical left-wing politics, and in the 1950s had been involved in the group running Socialist Review in London.[45] The LOGO Lab concept combined insights from Jean Piaget's and Lev Vygotsky's psychological studies of child development with the non-schooling principles of Ivan Illich.[46] It advocated an approach in which: 'the child programs the computer rather than the computer is being used to program the child.'[47] Papert also argued that the design of a programming

used for managing source code in programming projects. A conflict occurs when two or more programmers attempt to submit changes to the same section of code at the same time. Branches are a means of enabling programmers to work on copies of the code that have been 'branched off' into a separate development line from that of the main codebase, it can be used for testing out ideas before they are merged back into this. many aspects of the development of a software project can be traced in the records of a version control repository, making it a kind of discursive archive of how the software has been produced. For a discussion of these issues see: Simon Yuill, 'CVS' in Software Studies: A Lexicon, edited by Matthew Fuller, Cambridge MA: MIT Press, 2008.

[56]
Alasdair MacIntyre, After Virtue, third edition, University of Notre Dame Press: Notre Dame, Indiana, 2007.

[57]
Ibid., p.194.

[58]
Ibid., p.190-191.

[59]
Aristotle was a major influence on early Marx informing both his development of economic theory and his ideas on how a communist society might operate, particularly in terms of

language could reflect a particular political and ethical position. He criticised BASIC, another language originally designed for teaching programming, as demonstrating 'how a conservative social system appropriates and tries to neutralise a potentially revolutionary instrument.'[48] Although the Scratch Orchestra did not initiate from a defined political program, it nevertheless acted as a context for the development of a politicised arts practice informed by both Marxist and anarchist tendencies. It was through the Scratch Orchestra that Cardew was to acquire a profound political self-awareness, applying an explicit Maoist perspective to his own practice, and leading to his involvement in founding the Revolutionary Communist Party of Great Britain (Marxist-Leninist). Echoing Papert's criticisms of BASIC, Cardew similarly criticised the institutionalised conservativism of much music notation, demanding instead that 'all problems of notation will be solved by the masses.'[49] For both Papert and Cardew, pedagogy was a two way thing. The lab and the orchestra broke down distinctions between pupil and tutor, and placed learning in the context of self-directed production. In these ways they were forms of distributive practice.

Training in Contingency

An element of the contingent was essential to this. In Papert's eyes, one of the strengths of programming as a tool for learning, was the attitude to error that it encouraged. Encountering error, in the form of bugs, was an inevitable and necessary part of programming, especially that particular practice of programming developed at the AI Labs known as 'hacking'.[50] Papert pointed out that in conventional education, errors had a purely negative connotation. When a student makes a mistake they are discredited for it, losing marks or being punished, thereby encouraging a fear of error, leading to an unwillingness to stray from conventional boundaries and take risks. For the hacker, conversely, what mattered is not whether or not a mistake is made but rather how creatively it can be responded to. As with the Arkestra, embracing error is a productive possibility. The embracing of error is reflected in documents such as HAKMEM.[51] Short for 'hack memo', this was a collection of code snippets and programming ideas distributed amongst the hackers within the AI Labs – contributors include Richard Stallman, James Gosling and Marvin Minsky. Many of the entries utilise possibilities discovered through bugs

the capacities and potentials it might offer its citizens. See, for example, articles in: George E. McCarthy (editor), Marx and Aristotle: Nineteenth-century German Social Theory and Classical Antiquity, Rowan and Littlefield, Maryland: Savage, 1992.

[60]
MacIntyre, 2007, op. cit., p.191.

[61]
Self-actualisation, selbstbetatigung, was for Marx the opposite to alienated labour. It is an aspect of his thought that was informed by his readings of Aristotle.

[62]
Benjamin Franks, 'Anarchism and the Virtues', 2007, originally presented as 'Virtues and Anarchism: A non-essentialist account' at Alasdair MacIntyre's Revolutionary Aristotelianism: Ethics, Resistance and Utopia conference London Metropolitan University, June 2007. Franks also presents an outline of how MacIntyre's notion of virtue can be separated from previous essentialising models. As examples Franks gives the development of non-hierarchic organisational structures and forms of self-governance that emerged in the road protest camps of the mid-nineties and have evolved through the urban protest actions and social projects of the early 21st Century, in which hacklabs and FLOSS-based autonomous

and inconsistencies within the PDP computers that the AI Lab worked on. Other entries suggest ways that a particular algorithm might be played with, encouraging people to mess around with it in what can only be described as a form of aesthetic code play.[52] HAKMEM can be seen as the AI Lab's equivalent of the Scratchbooks exchanged between Scratch Orchestra members. Within the LOGO Labs, code was written and exchanged between students in a similar manner. Rather than planning out programs in advance, pupils would 'improvise' with their code responding to the how the turtle performed and modifying their programs accordingly. LOGO learning thereby operated through a similar feedback loop of coding-performing that livecoders such as Alex McLean identify as the basis of their practice and which builds upon the principle of the read-eval-print-loop.

Computers and programming languages present highly constrained environments that limit the possible varieties of interpretation that a particular notation may be subject to. The interpretation of notation by a human may be far less constrained. For Cardew this was a major concern in the development of new notations, for it presented both a danger and an opportunity. The opportunity was that notations need not only encode existing patterns or defined systems of sound, but could also be proposals and provocations to create new ones. The danger lay in the fact that a trained musician, when confronted with an unfamiliar notation system, rather than responding to it directly, might fall back into their personal predispositions and ingrained habits. The performance may simply become the regurgitation of old cliches and formulas like that of the amateur jazz musician described by Adorno, unable to stray from the existing models to which he has adapted and subordinated himself.[53] The trained musician approached a performance with a predefined system of producing sound against which the new notation was interpreted. What was novel in the new notation may simply be responded to as 'error' or noise within that system and therefore avoided. New notations required performers with a similar attitude to that of the hacker and LOGO Lab student, one who could respond creatively to the unknown and unexpected. The performer, therefore, could not rehearse such music but rather 'trained' for it like a martial art, developing ways of acting upon contingency.[54] This similarly developed through a feedback

media (such as Indymedia and free networks) have formed a significant part. These have been accompanied by a growing sense and application of historical narratives and self-critiques from within traditions of protest and political action such as Christopher Linebaugh's 'history from below', Bookchin's critique of 'lifestyle anarchism' (Murray Bookchin, Social Anarchism or Lifestyle Anarchism: An Unbridgeable Chasm, Seattle: AK Press, 1996) and Jo Freeman's 'Tyranny of Structurelessness' (Jo Freeman, 'The Tyranny of Structurelessness', 1970, http://flag.blackened.net/revolt/hist_texts/structurelessness.html). Similarly, Italian Autonomism re-articulated Marxist practice by drawing from the history of worker's action gathered in the Autonomia magazine and films such as Manuela Pellarin's Porto Marghera: The Last Firebrands, 2004. For a fuller discussion of 'history from below' in the context of British radical history see: Anthony Iles and Tom Roberts, All knees and elbows of susceptibility and refusal, http://caughtlearning.org/all_knees_and_elbows. Stewart Home's critique of 'Anarchist Integralism' is also important in this regard: 'Anarchist Integralism: Aesthetics, Politics and the Après-Garde', http://www.stewarthomesociety.org/ai.htm. Recent discussions of anarchist practice in the UK have largely moved away from the integrational model criticised by Home towards clearer forms of self-definition, see: Benjamin Franks, Rebel Alliances: the Means of British Anarchisms, Edinburgh: AK Press, 2007.

[63]
Richard Stallman, 'The GNU Project', Free Software, Free Society: Selected Essays of Richard M. Stallman, 2nd edition, GNU Press: Boston, 2004.

[64]
Richard Stallman, 'The GNU Manifesto', Free Software, Free Society: Selected Essays of Richard M. Stallman, 2nd edition, GNU Press: Boston,

loop of coding-performance that formed the basis of Scratch Music practice.

Through such feedback loops notation incorporates the experience of the contingent into future practice. What was the unexpected 'error' of the past becomes preparation for unknown future possibilities. In absorbing this a notation records the historical development of a practice, capturing different versions of how things could be done, and enabling comparison, analysis and synthesis of these. In both the LOGO Labs and Scratch Orchestra, this process of versioning was consciously engaged in, with the evolving knowledge, purposes and standards of the practitioner community acting as a form of version control identifying those practices that are most current and those which are conflicting or branching off.[55]

The Virtues of Practice

These examples emphasise practice over product. This is practice realised as more than just a set of techniques and skills however. It is practice that is consciously linked to, and helps define, particular practitioner communities: groups defined not by a common aesthetic, style, nor common collection of cultural references, therefore, but by commitments to shared practices. This socialised notion of practice parallels that outlined by Alasdair MacIntyre.[56] Whilst any practice may comprise of certain techniques, skills or activities, the practice itself is not determined solely through the performance of these. The activities of a given practice exist within a set of relations that are both social, in the relations between each practitioner and his or her contemporaries, and historical, in the relations of current activity in regard to an understanding of its past development, to how it has been practised in the past.[57] A practice may be judged in terms of its internal goods, those qualities and characteristics that enable it to flourish, and external goods, that which a practice produces which may become a property or possession of others who themselves are not practitioners.[58] Within a practice such as medicine, for example, an internal good may be the development of a new technique or understanding that enables doctors to realise more effective treatments, an external good would be the improved health of those patients who receive such treatment.

what does that mean???

2004, p.35.

[65]
Ibid., p.35.

[66]
Stallman, 'The GNU Project', 2004, op. cit., p.18.

[67]
See: Benjamin Franks, Rebel Alliances: the Means of British Anarchisms, Edinburgh: AK Press, 2007.

[68]
The four freedoms are listed in 'The GNU Project' but not in the General Public Licence itself.

[69]
Eric Raymond, The Cathedral and the Bazaar, http://catb.org/~esr/writings/cathedral-bazaar

[70]
Kropotkin's concept of 'mutual aid' is referred to in Raymond's How To Become A Hacker, http://catb.org/~esr/faqs/hacker-howto.html

[71]
MacIntyre, 2007, op. cit., p.74.

[72]
It would be wrong to suggest that Free Software developed through a conscious application of Marxist principles to software production, although Eblen Moglen for one has argued a conscious connection both to Marxist and left anarchist ideas of production: The dotcommunist Manifesto, http://emoglen.law.columbia.edu/publications/dcm.html, Anarchism Triumphant: Free Software and the Death of Copyright http://emoglen.law.columbia.edu/publications/anarchism.html

[73]
MacIntyre, 2007, op. cit., p.150.

[74]
For critiques of how Web 2.0 and 'long tail' capitalism has developed from Open Source models, see Dmytri Kleiner and Brian Wyrick, InfoEnclosure 2.0, Mute Vol.2 #4, Web 2.0 - Man's best friendster?, available online: http://www.metamute.org/en/InfoEnclosure-2.0, and Martin

MacIntyre's model of practice is central to his retrieval of Aristotelian 'virtue ethics', informed by Marxist social and economic analyses and developed as a critique and alternative to post-Enlightenment Liberalism and individualist ethics.[59] In MacIntyre's reading 'virtues' are those internal goods through which 'we define our relationships to those other people with whom we share the kind of purposes and standards which inform practices', and 'vices' those which inhibit or undermine that.[60] MacIntyre's notion of virtue is not a conservative one, virtues are not defined as a static table of tropes set down by institutions such as the church or state. In contrast to an ethics of duty based on obligation to a set of external standards to which the individual must aspire, virtue ethics arise from and are directed towards forms of practice. They are defined and realised through action rather than regulation or law and aim towards a general ethic of self-actualisation.[61] Different virtues may be open to change and development within the unfolding and evolution of a given practice. It is the practitioners who define that which is virtuous in regard to the aims of their practice. In applying MacIntyre's virtue ethics to contemporary anarchist practice, Benjamin Franks has emphasised this dialogic and immanent model of ethics that evolves through the interplay of practitioners and social situations: 'different virtues take priority in different contexts rather than conforming to a set of universal values'.[62]

The history of FLOSS, as given in the accounts of its formative practitioners, has very much been one of the evolution and discourse of practices. In 'The GNU Project', Richard Stallman writes about the MIT AI Lab as the first 'software-sharing community' in which building upon and adapting the code made available by others within the lab was the key basis through which ideas were developed and realised.[63] This is embodied in documents such as HAKMEM, and expressed in the form of a virtue ethic that echoes Aristotle: 'The fundamental act of friendship among programmers is the sharing of programs.'[64] Demonstrating how such a virtue ethic contrasts with that of regulative duty ethics, Stallman continues:

> …marketing arrangements now typically used essentially forbid programmers to treat others as

Hardie, FLOSS and the 'crisis': Foreigner in a Free Land?', Sarai Reader 04: Crisis/Media, edited by Shuddhabrata Sengupta, Monica Narula, et al. Delhi: Sara Media Lab, 2004, available online: http://www.sarai.net/publications/readers/04-crisis-media

[75]
Cornelius Cardew, 'Towards an Ethic of Improvisation', Treatise Handbook, London: Edition Peters, 1972, p.xvii.

[76]
Ibid., p.xx.

[77]
Ibid., p.xvii.

[78]
Ibid., p.xx.

[79]
Paolo Virno, A Grammar of the Multitude: For an Analysis of Contemporary Forms of Life, translated by Isabella Bertoletti, James Cascaito and Andrea Casson, Los Angeles: Semiotext(e), 2004.

[80]
Ibid., p.52.

[81]
Ibid., p.52.

[82]
Ibid., p.52.

[83]
Ibid., p.54.

[84]
MacIntyre, 2007, op. cit., p.197.

[85]
See Ken Thompson's contribution to An Oral History of Unix, http://www.princeton.edu/~hos/mike/transcripts/thompson.htm. This is also discussed in Martin Hardie, Time Machines and the Constitution of the Globe, http://openflows.org/%7Eauskadi/time;achines.pdf and http://auskadi.mjzhosting.com/ip.html

[86] The idea of performance is quite explicit in Papert's own account of the LOGO Labs, he cites the instruction methods of the Brazilian Samba Schools, as one model for how the labs worked.

[87]

> friends. The purchaser of software must choose between friendship and obeying the law.[65]

Within the practice of hacking, the sharing of code is an internal good. Stallman also relates the basic principles of hacking to an external good. In arguing against the ends orientated values of the 'proprietary-software social system' he proposes that the way in which software is made (its mode of production) is reflective of the 'kind of society we are allowed to have.'[66] Free Software hacking is therefore also a prefigurative practice in the sense outline by Franks, as it seeks to realise its ends within the means that achieve them.[67] When the 'proprietary-software social system' came into contact with the 'software-sharing community', the latter was brought into crisis due to the conflict of values that this provoked. This forced the need to explicitly define what were previously tacit values held by mutual consent, articulated by Stallman as the four freedoms of 'Free Software'. The four principles of Free Software can be seen as the articulation of a particular virtue ethic applicable to the production of software and the practice of programming. The fourth freedom specifically relates the internal good of hacking to an external good:

> You have freedom to distribute modified versions of the program, so that the community can benefit from your improvements.[68]

Eric Raymond's The Cathedral and the Bazaar develops its definition of 'Open Source' through a similar emphasis upon practice.[69] It appears that Raymond is also promoting a kind of virtue ethic that develops and articulates a particular practitioner community. The various references to Kropotkin's notions of 'mutual aid' and governance through 'the principle of common understanding' that are found in this and other of Raymond's writings would also suggests that he shares the kind of communitarian ethos of Stallman and one that might even relate to the 'practical anarchism' of Franks.[70] Raymond's approach, however, is fundamentally different. Whereas Stallman outlines a set of values appropriate to realising a form of socially-directed and self-actualised production, Raymond provides an analysis of how such production can be utilised for productive efficiency. In doing so he severs the relationship between the internal goods of hacking practice and the external

Coleman, 1982, op.cit, p.117.

[88]
Theodor W. Adorno, 'The schema of mass culture', The Culture Industry: Selected Essays on Mass Cuture, edited by J.M. Bernstein, London: Routledge, 1991, p.76.

[89]
Quoted in Szwed, 2000, op.cit., p.112.

[90]
Quoted in Szwed, 2000, op.cit., p.241. Ihnfinity Inc had grown out of an earlier discussion group that had been setup alongside the Arkestra to explore aspects of mythology and science.

[91]
See: Looker, 2004, op.cit.

[92]
Anthony Braxton quoted in Lock, 1988, op.cit., p.317.

[93]
Some of these problems are discussed by Eley in Cardew and Eley, 1974, op. cit. Hanne Boemisch's film, Journey To The North Pole (1972) documenting the Scratch Orchestra tour of Northern English towns, appears to illustrate some of this gulf and the growing tensions that were emerging in the group.

[94]
Testimonies from different Scratch Orchestra members are given in Twenty Five Years From Scratch, op.cit., many of these comment on the issues surrounding the groups' break-up.

[95]
Eddie Prevost in Twenty Five Years From Scratch, 1994, op.cit., p.38.

[96]
Michael Chant, 'A Turning Point in Music History', in Experimental Music Catalogue, http://www.users.waitrose.com/~chobbs/chant.html

[97]
Cornelius Cardew, 'Treatise: Working notes', Treatise Handbook, London: Edition Peters, 1972, p.iv.

[98]

goods of communitarian production that are the basis of Free Software. The virtues of Free Software are replaced by the rules of Open Source – Raymond literally defines 19 rules of Open Source production. In place of an ethics of production we are presented with a management model which, according to MacIntyre, is antithetical to virtue, aiming only towards 'the most efficient means of achieving whatever is proposed.'[71] Whilst both Free Software and Open Source offer models of production that are collaborative they differ fundamentally in how this is orientated. Free Software presents a model of collaboration that is distributive, it seeks to enable others to have disposition over their own production.[72] MacIntyre would argue that this demands an ongoing process of critical judgement and the 'exercise of the virtues' appropriate to such a practice which cannot be subject to a 'routinizable application of rules.'[73] Open Source, on the other hand, presents a model of collaboration that is acquisitive, it seeks to harness the labour of others so as to reduce production costs and increase profits (reducing liability is often identified as a key saving within commercial Open Source projects), or create profits in previously unrecognised areas. This can be seen in tracing the evolution of Open Source style licensing and production models away from a set of positive freedoms enabling self-disposition towards a set of negative freedoms acting upon a liberalised sharing economy. These are exemplified in the variations of the Creative Commons licenses and the regulative, aspirational (rather than virtuous) sharing of Web 2.0.[74]

As with Free Software, the history of the Scratch Orchestra can be understood as one of a particular practitioner community evolving its own ethics of practice. The constitution itself defines the group in terms of the activities that it will pursue and develop through. That the constitution was subject to rewriting and revision during the time of the group's existence indicates there was an ongoing evaluation of this definition in relation to that evolving practice. One of the texts Cardew wrote in the period leading up to the formation of the Scratch Orchestra suggests ways in which the practices of the Orchestra might be understood in relation to a conscious form of virtue ethic. The essay is titled 'Towards an Ethic of Improvisation' and opens with the sentence: 'I am trying to think of the various kinds of virtue or strength that can be developed by the musician.'[75] It ends with an outline of seven 'virtues that a

Theodor W. Adorno, Sound Figures, translated by Rodney Livingstone, Stanford: Stanford University Press, 1999 p.214.

[99]
The creation and impact of the Scratch Orchestra Ideological Group can be seen as the creation of a political vanguard within the Orchestra, in accordance with the kind of orthodox Leninist-Maoist politics of those who set it up. Benjamin Franks' critique of vanguardism highlights many of the problems that the Orchestra experienced through this, see: Benjamin Franks, 'Paternalism and Vanguardism' presented at the Civil Rights, Liberties and Disobedience conference, Centre for the Study of International Governance, Loughborough University, July 2007.

[100]
Robert Nozick, Anarchy, State and Utopia, London: Blackwell, 1978.

[101]
Richard Stallman, http://www.gnu.org/philosophy/rms-comment-longs-article.html

[102]
Robert T. Long, 'The Libertarian Case Against Intellectual Property Rights', originally published in the Autumn 1995 issue of Formulations, available online: http://libertariannation.org/a/f31l1.html

[103]
The Oekonux mailing list has provided a useful, and sometimes contentious, forum for the discussion of such issues: http://www.oekonux.org

[104]
Henri Pousseur, quoted in Eco, 2004, op.c it., p.168.

[105]
Ibid., p.168.

[106]
Ibid., p.174.

[107]
Virno, 2004, op. cit., p.56.

[108]
Martin Hardie, 'The Factory Without Walls', http://openflows.org/~auskadi/factorywoutwalls.pdf, see also:

musician can develop', these include 'simplicity', 'selflessness', and 'preparedness'. The virtue of 'forbearance' is described in terms that echo something of Sun Ra's attitude: 'Overcoming your instinctual revulsion against whatever is out of tune (in the broadest sense).'[76] One of the most significant aspects of the essay is its emphasis upon improvisation as a form of 'active life'. It is in this that it connects most strongly with the later activities of the Scratch Orchestra and in particular their stated aim to 'function in the public sphere'. Virtue, Cardew tells us, 'is viewed to best advantage in action'[77], whilst improvisation is only purposeful when 'it occurs in a public environment' for 'its force depends to some extent on public response.'[78] Improvisation, like virtue, depends on a social context and both have value only when realised through actions within such a context. It is on this basis, as Paulo Virno explores, that improvisation exemplifies virtuosity.

The Praxis of Virtues

Like MacIntyre, Virno's exploration of virtuosity derives from a reading of Aristotle via Marx.[79] Virno defines virtuosity in terms of two particular qualities. The first is that of 'an activity which finds its own fulfilment (that is, its own purpose) in itself' and therefore has no end product and, like improvisation, no 'object which would survive the performance.'[80] The second quality is that it is 'an activity which requires the presence of others, which exists only in the presence of an audience.'[81] For Virno, this relates virtuosity to Aristotle's notions of political action, to praxis rather than poesis. Poesis aims towards the making of an end product 'separated from action', whereas in praxis action is an end in itself.[82] This in turn is related to Marx's distinction between an 'activity-with-end-product', such as conventional manufacture, and an 'activity-without-end-product', such as that of the performer, the waiter, the teacher, and the medical doctor.[83] Virno argues that such 'activity-without-end-product' is a poesis, a way of making, that tends towards the condition of praxis. For Aristotle, the action which finds fulfilment in itself is also the virtuous action, and following from this, MacIntyre describes those who pursue a practice in terms of furthering its internal goods as those who similarly find fulfilment in the activity itself.[84] Virtuosity then could be defined as 'the performance of a practice at the height of its virtues' and a form

Martin Hardie, 'Change of the Century: Free Software and the Positive Possibility', Mute Vol. 2 #1, Underneath the Knowledge Commons, available online: http://www.metamute.org/en/Change-of-the-Century-Free-Software-and-the-Positive-Possibility

[109]
Virno, 2004, op. cit., p.62.

[110]
See for example: Brian Ashton, 'Factory Without Walls', MUTE vol. 2 issue 4, Web 2.0 - Man's best friendster?, available online: http://www.metamute.org/en/Factory-Without-Walls. Regus are one of the largest 'instant offices' providers offering services ranging from actual office spaces that can be rented by the hour to receptionists and 'on demand' call centres: http://www.regus.com

[111]
Virno, 2004, op. cit., p.66.

[112]
Virno, 2004, op. cit., p.85.

[113]
Bourriaud, 2002, op. cit., p.30.

[114]
For an analysis of call centre work see: Kolinko, Hotlines: Call Centre Inquiry Communism, Oberhausen: Kolinko, 2002. Aufheben use call centre work in illustrating some of the problems and limitations of 'immaterial labour' as outlined by Virno, see Aufheben, 'The Language of Retreat: Paulo Virno's A Grammar of the Multitude', Aufheben, issue 16, 2008, pp.36 - 48.

[115]
For Cardew's criticisms of 'randomness' and 'confusion' as avant-garde strategies see: Cardew and Eley, 1974, op. cit., p.45 and p.77. For critiques of these aspects of post-modernism can see: Frederic Jameson, Postmodernism: Or, the Cultural Logic of Late Capitalism, London:Verso, 1992, and Michael Hardt, Anthony Negri, Empire, new edition, Cambridge MA: Harvard University Press, 2001.

[116]
This is a key aspect of Virno's 'exit strategy', see: Virno 2005,

of poesis that is realised as praxis. This is clearly exemplified in Cardew's ethics of improvisation and carries through into the Scratch Orchestra as the conscious creation of a practitioner community based around such an ethic.

This can also be seen to apply to hacking, which similarly demonstrates how a form of production-through-notation may relate such virtuosity to an ethic of distributiveness. Whereas commercial software production emphasises the creation of distinct software products, hacking emphasises code as part of a ongoing dialogue between practitioners. In the accounts of the UNIX oral history project, Ken Thompson, one of the developers of UNIX, recalls his surprise at seeing how Bell's marketing people took the UNIX operating system which to him was 'part of a continuum' that could be adapted and extended as required, and packaged it as a discrete product to be consumed as a fixed entity.[85] In the LOGO Labs coding was pursued as a means of enquiry that found satisfaction in itself but which was directed towards collective dialogue between students and through the performance of the turtle.[86] Similarly, FLOSS projects today are primarily presented through their code repositories which foreground the project as a continuum of production and act as the 'public' context in which the activity of hacking finds an audience. Commercial software production is acquisitive in that firstly it acquires the labour of others, that is then sealed under employment contracts and copyright, and secondly demands that it is consumed as an acquisition whose disposition is similarly restricted. Copyright became significant to the emergence of commercial software as it is the application of copyright, used in its conventional restrictive sense, that is used to define the code as a fixed product. Free Software, in contrast, emphasises the code as something that enters into a continuum of production. So whilst there is an 'output' in the form of written code, it enters into circulation in a way that is distinct from a conventional product. Free Software is an 'activity-without-end-product' not in the sense of having no output, but rather in the sense of constantly creating the capacity for production elsewhere. The fact that the knowledge of production can be expressed in notation, in the form of source code, is integral to this. This is echoed in the Scratch Orchestra with its emphasis upon the production of notation as both an ongoing and public activity. The notationally based improvisations of the Scratch Orchestra

op. cit., p.70. For a critique of Virno and the relation of his exit strategy to 'entrepreneurial Autonomism' see Aufheben, 2008, op. cit., p.31.

[117]
Aufheben, 2008, p.29. The phrase is used by Aufheben in criticising Virno's concept of multitude rather than applied to Creative Commons. The conflicts between free-as-in-libre and free-as-in-unpaid labour are discussed in greater detail in Tiziana Terranova, 'Free Labor: Producing Culture for the Digital Economy', Electronic Book Review, 2003, available online: http://www.electronicbookreview.com/thread/technocapitalism/voluntary

[118]
The 'mail must get through' was one of Raymond's imperatives for Open Source. These forms of communications enterprise have been described by McKenzie Wark as 'vector capitalism', see: McKenzie Wark, A Hacker Manifesto, Cambridge MA:Harvard University Press, 2004, available online: http://subsol.c3.hu/subsol_2/contributors0/warktext.html. In regard to the use of such forums for radical dissemination, such as anarchist film archives on YouTube, or activist self-organisation through Facebook, the question remains as to whether these are unwittingly accommodating themselves through their own contribution to vectoralist commodification or are they exploiting the 'white noise' of Web 2.0 to spread content that might otherwise be blocked or more overtly appropriated?

are therefore significantly different from those performed without a score. For whilst the actual performance itself may never be repeated the capacity for its production elsewhere remains. Notation therefore, not only contains the possibility of retaining the history of how a practice develops, thereby aiding its development towards its own internal goods, but also of enabling those internal goods to be expressed in a form that creates capacity for others, thereby becoming external goods.

Black Notated Music

How a notation comes to be defined and how it is distributed are inherently political issues. This distribution extends beyond the publication of music scores and software code such as addressed through the copyleft mechanisms used by the Scratch Orchestra and FLOSS. As Ornette Coleman recalls, the very visibility of notation within the production process, how it is revealed and concealed, is itself dependent upon and expressive of particular relations of power and political context:

> I once heard Eubie Blake say that when he was playing in black bands for white audiences, during the time when segregation was strong, that the musicians had to go on stage without any written music. The musicians would be backstage, look at the music, then leave the music there and go out and play it. He was saying that they had a more saleable appeal if they pretended to not know what they were doing. The white audience felt safer.[87]

The denial of notation described in this episode is a denial of the black musician's self-legitimation. If the use of a notation may provide the basis for transcribing and re-coding the development of a practice, its own history of making and reflection upon that, then the denial of notation is a denial of such history and therefore a denial of the practitioner's basis for legitimation. It is from this perspective that Coleman distances his own practice from the idea of improvisation, for this form of 'virtuosity' became the basis of a denial of legitimation. The 'free playing' that he and other black jazz musicians promoted in the 1960s was not simply free in the sense of a break from conventional musical structure, but also free in breaking away

from the condition of being 'improvisers in a compulsory situation.'[88] This led to the development of new performance venues, many situated directly within black communities, and of the conscious articulation of practice as a form of research. Lester Bowie of the Art Ensemble of Chicago adopted a scientist's white lab coat on stage to announce the performance itself as a site of radical experiment. As Sun Ra encouraged his Arkestra: 'You're not musicians, you're tone scientists'.[89] Ra followed this concept further through the creation in 1967 of Ihnfinity Inc, a research corporation intended 'to own and operate all kinds of research laboratories, studios, electronic equipment, electrochemical communicational devices of our own design and creativity...'[90] In St. Louis the Black Artists' Group set up a Training Centre to create a discussion forum for the local community that alongside performances, rehearsals, and workshops, hosted regular meetings and debates about local issues.[91] For Anthony Braxton the relation of notation to legitimation became the basis of research that has been the focus of his work ever since, the development of what he calls 'Black Notated Music'. 'Black Notated Music' goes beyond the simple description of sounds on a page and engages with the extended functionality of sound at a socially structuring level: 'notation can be viewed as a factor for establishing the reality platform of the music.'[92]

Whilst on the surface these may appear to mirror the pedagogic basis of projects like the Scratch Orchestra and LOGO Labs, they developed from an entirely different trajectory. Although the pedagogics of Cardew and Papert aimed, on the one hand, to break down certain established social structures determining acquisition of skills in music and programming, pedagogy was also the basis upon which they integrated their work back into existing institutional frameworks, thereby legitimating it in the terms of those institutional values. In particular this legitimated their 'non-commercial' status. A similar case could be made for Free Software's dependency on academia, and suggests a potential area of conflict of interest within artist-run workshops, or at least highlights the tensions under which self-valorising labour is forced to 'pay the rent'. For black musicians in the USA of the 1960s, for whom even basic access to education was an issue, such avenues were not available. The appropriation of 'white' lab coats and research culture did not seek accommodation within such institutions but rather questioned their very use as legitimising mechanisms. Eventually the Scratch Orchestra was to become aware of its own dependency on such external forms of

legitimisation and the 'compulsory situation' within which it operated.

Instrumentalising the Collective

In 1972 tensions began to emerge within the Scratch Orchestra. It was felt by some that the group was operating in a fashion that was becoming contradictory with its aims and a 'discontents file' was set up into which people could address these grievances.[93] In response, Cardew, Keith Rowe and John Tilbury established a Scratch Orchestra Ideological Group applying a practice of Maoist self-critique amongst the Orchestra members. Whilst a process of self-criticism within the Orchestra may have been beneficial, this approach merely exacerbated the situation. Many felt that it was the imposition of one self-appointed elite exerting its authority over the Orchestra as a whole and that the Ideological Group's dismissal of certain initiatives from other members did not properly recognise their own political basis.[94] Rather than finding a new clarity of purpose, the Orchestra fell apart. As one member, Eddie Prevost, was to later comment, the fundamental contradiction confronting the Orchestra was perhaps its dependency upon its own constitution 'legislating for noncomformity'.[95] Another member, Michael Chant, observed that the constitution was itself a 'score'.[96] The Orchestra was then the product of this score, a score that carried the name of only one author: Cornelius Cardew. From this perspective the setting up of the Scratch Ideological Group might be seen as an attempt to re-assert authorship over Cardew's 'composition', echoing the concern of his earlier writings that 'the score must govern the music'.[97] This may be a classic example of an ideological vanguard acquiring and instrumentalising the collective for its own ends, and the rebirth of the author in a group attempting to move beyond such notions of singular authorship. In refusing to succumb to such ideological and authorial acquisition, a necessary restructuring of the 'composition' of the Orchestra was taking place. The inherently distributive quality of the Orchestra empowered forms of self-actualisation that rendered the need for a single cohering group unnecessary. Many members went on to continue in different practices that extended the radical praxis that had developed within it. The breakup, therefore, represented not the failure of its members, but rather the breaking of the limit between the formal structure of the score/constitution and the people who were the 'substance' of the

Orchestra. In words that Adorno used to describe an error of notation in one of Schoenberg's serial compositions, this represented

> […] the breakthrough of the substance to be structured, the point where it encounters the structuring process and but for which the latter could not be legitimated.[98]

The imposition of ideological judgement upon the group may have had a similar effect as the recuperation of Free Software practice under the managerial aims of Open Source, undermining the evolution of the practice under its own internal good, and acting as an acquisitive force that separates the practice from the realisation of its accordant external goods.[99]

Legislating for Nonconformity

There are parallels with Free Software's current reliance on copyleft and the GPL which can also be seen as a way of 'legislating for noncomformity'. The GPL may 'reverse' the normal restrictions created by conventional copyright, but it nevertheless depends upon their basic legal framework, and therefore upon a legalised notion of freedom that is realised through property ownership. Hence the attraction of copyleft for right-Libertarians such as Raymond. Indeed it may be argued that copyleft, as it is currently realised, rather than embodying a form of 'production in common' actually exemplifies something closer to Robert Nozick's 'just transaction'.[100] The problem with copyleft in its current form, and the notions of 'remix' culture and legalised 'appropriation' culture that have been developed from it, are that they merely present an alternative within proprietary, acquisitive production (capital) rather than an alternative to that. This is echoed in the active promotion of Jeffersonian 'liberty' amongst advocates of Open Source and Creative Commons such as Eric Raymond and Lawrence Lessig. To place an emphasis upon copyleft as an end in itself, and upon the GPL as the key defining document of Free Software, is therefore potentially contrary to the aims of Free Software. This is borne out in a comment from Stallman:

> Free software is a matter of freedom. From our point of view, precisely which legal mechanism is used to deny

> software users their freedom is just an implementation detail. Whether it is done with copyright, with contracts, or in some other way, it is wrong to deny the public the freedoms necessary to form a community and cooperate. This is why it is inaccurate to understand the Free Software Movement as specifically a matter of opposition to copyright on software. It is both more and less than that.[101]

It is significant that this was given in response to Robert T. Long's promotion of copyleft as appropriate to the values of a right-Libertarian free market.[102] It is perhaps best to view the GPL and copyleft as tactics affording certain leverage in current circumstances therefore, and the proliferation of 'open' licences in recent years might be more a sign of the accommodation of resistant practices to an order of legitimation that they might best avoid, for under current law there is no magic licensing scheme that will bring an end to proprietary production.[103]

[101] Richard Stallman, http://www.gnu.org/philosophy/rms-comment-longs-article.html

[102] Robert T. Long, 'The Libertarian Case Against Intellectual Property Rights', originally published in the Autumn 1995 issue of Formulations, available online: http://libertariannation.org/a/f31l1.html

[103] The Oekonux mailing list has provided a useful, and sometimes contentious, forum for the discussion of such issues: http://www.oekonux.org

[104] Henri Pousseur, quoted in Eco, 2004, op.c it., p.168.

[105] Ibid., p.168.

Distributive Production

The conflicts within the Scratch Orchestra and the conflicts between Free Software and Open Source illustrate the distinctions within forms of production between those that are collective and distributive, and those that are collaborative and acquisitive. A distributive practice enables the disposition of labour by others under their own direction, whilst an acquisitive one accumulates the labour of others without regard to their self-disposition. It also exposes the conflict that can emerge when a practice that has developed within a self-constituent community becomes subject to external forms of constitution and legitimation. Not all collaboration is inherently distributive, therefore. The nature of the power relations within it, and the disposition and legitimation of production they enable, may be subject to forces that operate in opposing ways.

> It is not out of the question that we consider these notations as a marketable product.[104]

So wrote the composer Henri Pousseur in a description of his composition Scambi, composed in 1957, and presented as a key example in Eco's study of the open work. 'Scambi is not so much a musical composition as a field of possibilities', Pousseur explains, 'an explicit

invitation to exercise choice.'[105] His language anticipates that of Web 2.0 and the liberal market place in which, to use Eco's words, openness is 'the fundamental possibility of the contemporary artist or consumer.'[106] Scambi predicts the notions of personalised commodity and networked production in which the distinction between producer and consumer is diminished, not in a form that extends free disposition over capacities of creation, but rather operates acquisitively on the collaboration of the consumer. In some ways it points towards the legacy of Papert's 'potentially revolutionary instrument' becoming part of a consumer toy range in LEGO Mindstorms. Scambi provides an early example of how, according to Virno: 'Virtuosity becomes labour for the masses with the onset of a culture industry.'[107] The transformation from the factory-based production of the Ford era to the network-based production of the post-Fordist era that Virno addresses, is a transformation in the notation of production in general. All notations of production are inherently architectural for they all inscribe and interweave relations of power. This can be expressed in the sense of the archi-tectural as residing in an etymological family that links to terms such as hierarchy, monarchy and anarchy on the one hand and textuality and textile on the other. A notation proposes, and is taken up within, particular architectures of production and inscriptions of power. The history of notation is therefore integral to the history of the factory, the space in which production is physically marked out and performed. The significance of groups such as the Scratch Orchestra in the late 1960s to the emergence, nearly forty years later, of livecoding and a revival of interest in collective improvisation, can be related to the transition from the singular, coherent factory-within-walls of Fordist production-line manufacture to the polymorphic, unstable factory-without-walls of post-Fordist networked manufacture. As Martin Hardie argues, it is UNIX, with its networked, distributed filesystem, that created the basic notational inscription of the factory-without-walls.[108] Where once, Marx compared the factory manager to the conductor of a classical Orchestra, rehearsing a score set in machine and stone, now 'the tasks of a worker or of a clerk no longer involve the completion of a single particular assignment, but the changing and intensifying of social cooperation.'[109] The factory has become an improvised collective ensemble composed of temporary contract workers, outsourced partners, 'instant office' providers, and consumers who are not even aware they are contributing labour to its production.[110] All performing what Virno describes as 'virtuosity without a script.'[111]

[106]
Ibid., p.174.

[107]
Virno, 2004, op. cit., p.56.

[108]
Martin Hardie, 'The Factory Without Walls', http://openflows.org/~auskadi/factorywout
walls.pdf, see also: Martin Hardie, 'Change of the Century: Free Software and the Positive Possibility', Mute Vol. 2 #1, Underneath the Knowledge Commons, available online: http://www.metamute.org/en/Change-of-the-Century-Free-Software-and-the-Positive-Possibility

[109]
Virno, 2004, op. cit., p.62.

[110]
See for example: Brian Ashton, 'Factory Without Walls', MUTE vol. 2 issue 4, Web 2.0 - Man's best friendster?, available online: http://www.metamute.org/en/Factory-Without-Walls. Regus are one of the largest 'instant offices' providers offering services ranging from actual office spaces that can be rented by the hour to

In the unstable environment of post-Fordist production, producers and consumers are caught in a condition of perpetual contingency. The agile responsiveness of the virtuoso hacker becomes the basic skill of the average employee:

> Only one who is experienced in the haphazard changing nature of the forms of urban life knows how to behave in the just in time factories.[112]

The social-networked pro-sumer becomes a catalyst to the combinatorial logic of late capitalist production, feeding the permutational offerings of personalised commodities and productised services that, in accordance with Bourriaud's aesthetic:

> operate like a relational device… a machine provoking and managing individual and group encounters.[113]

Virtuosity under post-Fordism compels us all to become 'improvisers in a compulsory situation.' This is virtuosity without virtue however. It directs practice towards 'external goods' set by managerial goals rather than arising from the 'internal goods' of those practices themselves. Collaboration becomes the dominant paradigm both of managerial control and everyday consumption. It constructs collaboration through relational mechanisms that are acquisitive rather than distributive. Contrary to Virno's claim, however, we are not performers without a script but rather enmeshed in endless small scripts and programs. Every aspect of our lives is notated to a degree not previously known and we are constantly challenged by new scores and scripts that we must perform in order to complete even the most mediocre task. It is through such notation that immaterial labour is valorised and managed. This is exemplified in the call centre worker who is the emblematic counterpart to the livecoder, performing actions and words composed in scripts and programs orchestrated on the computer screen they work from. These conduct a performance in strict tempo dictating duration of tasks and work breaks, a virtuosity like that of Sheherazade that must constantly justify and renew itself, trapped in an endless read-eval-print-loop. Through the interrupt mechanism of cold calling, this performance draws its audience into the collaborative labour of data acquisition, marketing surveys and sales support.[114]

receptionists and 'on demand' call centres: http://www.regus.com

[111] Virno, 2004, op. cit., p.66.

[112] Virno, 2004, op. cit., p.85.

[113] Bourriaud, 2002, op. cit., p.30.

[114] For an analysis of call centre work see: Kolinko, Hotlines: Call Centre Inquiry Communism, Oberhausen: Kolinko, 2002. Aufheben use call centre work in illustrating some of the problems and limitations of 'immaterial labour' as outlined by Virno, see Aufheben, 'The Language of Retreat: Paulo Virno's A Grammar of the Multitude', Aufheben, issue 16, 2008, pp.36 - 48.

[115] For Cardew's criticisms of 'randomness' and 'confusion' as avant-garde strategies see: Cardew and Eley, 1974, op. cit., p.45 and p.77. For critiques of these aspects of post-modernism can see: Frederic Jameson, Postmodernism: Or, the Cultural Logic of Late Capitalism, London:Verso, 1992, and

The scores and scripts of these performances have been hidden and we are unable to narrate and legitimise our own actions withen them. In a wider sense, therefore, livecoding is emblematic of the demand that the scores be brought on stage, for only then can the problems of notation be properly addressed. Under a regime of acquisitive inscription, however, we may need to reverse Cardew's proposition: the problems of notation should not be solved by the masses, but rather the notations of production must be made constantly problematic.

In pursuing such a tactic we should be careful, as Cardew once warned, not to fall back on avant-gardist cliches of simply creating random noise and confusion as an end in itself, or into the spiralling solipsisms of post-modernism which have done so much to shape and inform the rhetoric and forms of personalised commodity culture.[115] Similarly, we should be wary in following Virno's call for 'unrestrained invention' that it may simply be the corollary of avant-gardist randomness, a permutational generator of lifestyle commodities and niche markets.[116] If Free Software, and related practices, are intended to realise a form of free-as-in-libre labour rather than free-as-in-unpaid-expropriated labour this can only happen in foregrounding and realising freedoms of production, rather than the 'bourgeois freedoms of circulation' promoted by Creative Commons and remix culture.[117] As enterprises such as Facebook and YouTube demonstrate, profits do not fall as the signal-to-noise ration of communication increases. Any noise, any unrestrained invention, that can be acquisitively channelled can be commodified. In such arenas it is circulation not content that counts, and as long as the 'mail gets through' the message is irrelevant.[118] In questioning a form of production in terms of its distributiveness we are asking questions as to how capacities and freedoms of production are articulated and enabled in relation to circulation, rather than cohering and channelling supposedly 'autonomous' labour under acquisitive 'collaboration' models. We should be careful, however, not to valorise distributiveness as an end in itself, for this would bring about a similar severing or misdirection between internal and external goods, and between means and ends, such as we see under the current conditions of 'collaborative' production.

Michael Hardt, Anthony Negri, Empire, new edition, Cambridge MA: Harvard University Press, 2001.

[116]
This is a key aspect of Virno's 'exit strategy', see: Virno 2005, op. cit., p.70. For a critique of Virno and the relation of his exit strategy to 'entrepreneurial Autonomism' see Aufheben, 2008, op. cit., p.31.

[117]
Aufheben, 2008, p.29. The phrase is used by Aufheben in criticising Virno's concept of multitude rather than applied to Creative Commons. The conflicts between free-as-in-libre and free-as-in-unpaid labour are discussed in greater detail in Tiziana Terranova, 'Free Labor: Producing Culture for the Digital Economy', Electronic Book Review, 2003, available online: http://www.electronicbookreview.com/thread/technocapitalism/voluntary

[118]
The 'mail must get through' was one of Raymond's imperatives for Open Source. These forms of communications enterprise have been described by McKenzie Wark as 'vector capitalism', see: McKenzie Wark, A Hacker Manifesto, Cambridge MA:Harvard University Press, 2004, available online: http://subsol.c3.hu/subsol_2/contributors0/warktext.html.
In regard to the use of such forums for radical dissemination, such as anarchist film archives on YouTube, or activist self-organisation through Facebook, the question remains as to whether these are unwittingly accommodating themselves through their own contribution to vectoralist commodification or are they exploiting the 'white noise' of Web 2.0 to spread content that might otherwise be blocked or more overtly appropriated?