**Chapter 1 and 2 of thesis including selected appendices**

# MODELS OF OPEN SOURCE PRODUCTION COMPARED TO PARTICIPATIVE SYSTEMS IN NEW MEDIA ART

DOMINIC SMITH

PhD 2011

# Abstract

The term ‘Open Source’ has in the past decade been used very loosely in relation to art and social practices. This research compares the production processes of Open Source software production with those of participative new media art projects.  The contextual review examines the behaviours of computer scientists from the 1960s onwards, including hacking, interaction over computer networks and shared use of computers when they were a scarce resource. Collaborative environment strategies for personal success are traced onto free software, FLOSS (Free Libre Open Source Software) and open source. Licencing and copyright are examined in relation to distribution. The development of participative art projects is also traced, in relation to new media, and these ethics of authorship, freedom, sharing and distribution.

The research compares certain political and social ethics between software and art. It identifies different levels ‘openness’ and different kinds of hierarchies within production systems, including hierarchies of skill, approval, gatekeeping, and time. Interviews with key open source practitioners help to identify these hierarchies.

As part of a practical body of research, a series of participative projects were developed by the researcher. These included both online and physical space participation, including the *Random Information Exchange* series and *Shredder*. These were designed to test the various principles of open source within a new media art context. Through the successes and limitations of these projects, the elements of a project that are necessary for it to be to be classed as open source were identified. The findings of the research describe important differences in hierarchical structures of projects’ production and distribution, and identify key elements including the ‘ownership’ of projects over time, and the importance of differentiating the ‘instigator’ role from the ‘developer’ role.

# Acknowledgements

Firstly and foremost I would like to thank Professor Beryl Graham and Dr. Sarah Cook for their continuous support throughout this research. Their enthusiasm, encouragement and invaluable input have made this research possible. It has been an honour to be their student.

I wish to thank friends, colleagues and anyone who has participated in this research. I would like to thank my collaborator at The Polytechnic, Sneha Solanki for her feedback and support throughout this research.

Thanks to my parents Joan and James for encouraging me to follow my passions in life despite the odds. I am thankful to my children Isaac and Ruby for reminding me to have fun.

Lastly I would like to thank Johanna to whom I dedicate this thesis for her enduring support, patience and belief that I could do this.

# Chapter 1: Introduction

## *1.1 Starting point of research*

In the early stages of my career I ran arts workshops in education and managed a studio group that curated a number of small shows and a film and new media festival. This led to a position within Tyne & Wear Museums & Archives first as an assistant curator, which was followed by a series of positions as part of the outreach team. Due to my increasing involvement with new media and digital technologies as part of my artistic practice, I was often called upon to be part of outreach projects that involved the use of these technologies. One project was named Museums Outreach Online (MOO). This project had a base of Windows desktops that members of the public could use to access the Internet. It also had a number of laptops that could be taken into the local community to develop digital projects. However, it became apparent after delivering a number of these projects that this process was flawed. Laptops fitted with expensive licensed software were being taken to community groups which were trained in their use and then just as users were gaining competency the project would end and all resources would be removed leaving the users in the same disadvantaged position they were in before the project began. This was at a time when the average price of a laptop was £1000, adding that to the cost of multimedia software such as Photoshop, Flash and Director made purchases amongst the groups the project was working with overly prohibitive. In 2003, I attended a Pure Data training course at Montevideo in Amsterdam. Upon arriving, the workshop leaders erased Windows from my laptop and installed Linux as they claimed it had the largest number of useful Pure Data extras. This was my first real introduction to Linux and the world of free operating systems and open source software and it took me from my misguided belief that I was an expert computer user and placed me in the position of a novice where I had to genuinely gain an understanding of how the operating system interacted with the computer in order to use it productively. On returning to England, it was apparent that a combination of open source software and recycled technology could provide a solution to the earlier problem of removing the resources from the groups I had been working with by massively reducing the cost and further encouraging knowledge transfer. I then went on to form The Polytechnic with a number of other like-minded artists and went on to participate in a national scheme titled *Grow Your Own Media Lab* where the group was partnered with a number of organisations to develop new media labs and projects with members of the wider community where the resources and knowledge would remain after the projects' end (see Appendix A).

At the same time, a growing number of other arts projects in other parts of the world were also experimenting with 'open source' methods, but definitions of these methods were very diverse or inaccurate, and there seemed to be little sharing of practical knowledge or experience. The research therefore started from a ‘felt need’ to examine open source methods, and to critically reflect on the practice of how these ways of working compare with other participative new media art projects.

## *1.2 Research aims*

My research explores models of open source production compared to participative systems in new media art. My research aim is to examine and compare modes of open source software production with emerging modes of collaborative and participative organisation within the new media art world, including an analysis of practical project experience.

The examination of the field aims to accurately identify different kinds of open source practice, in order to be able to compare the similarities and differences between open source methods and participative new media art methods. The comparison aims to inform my peers working in similar fields through an accessible body of work.

A narrative of the research process undertaken to acheive this aim is described in section 1.6, including the roles of the practical bodies of work.

## *1.3 Methodology*

To explore the open source development process it was necessary to create work with and participate in the development communities. Because the starting point of the research came from practice, I chose a primarily practice-led action research methodology. Like many other practice-led PhDs, a contextual review of the field informs a series of practical projects, with reflection on each project informing the intent of the next project (Akam 2004, Bunnell 1998). In the case of my projects, there were sometimes several iterations of the same basic model, such as the differing iterations in different venues of the *Random Information Exchange*.

In reflecting on the series of practical projects, the method does not attempt to measure ‘success’ or technical detail, but rather to identify kinds of open source method, and to reflect on which methods had what effects on the outcomes of each project, particularly concerning patterns of participation or otherwise. Each project leads on to the next, following cycles of enquiry which attempt to address the failings of the last and capitalise upon successes. This was part of an action research methodology that is best described by Heron and Reason.

> "There is explicit attention, through agreed procedures, to the validity of the inquiry and its findings. The primary procedure is to use inquiry cycles, moving several times between reflection and action." (2006, p.144).

As the research is concerned with the processes of software development along with production methods, I have looked closely at Casey Reas and Ben Fry's process of development for processing (an open source programming language and software development environment) (2007) and Chun Lee's method of making sound art software in response to a contextual review of open source practices (2008). By developing work that encouraged collaborative development of a critically engaged artwork, my method differs slightly from that of artists and researchers who develop a tool or system which is then shared in order to gather data.

Reas used the development of processing to gather research about the impact of open tools for artists. His method was to develop this environment and observe its use in the wild.  A similarity with my research method is that at the time of development, Fry and Reas did not know what the scale of uptake would be, how it would be used or how the community would contribute to its development, i.e. write libraries, make suggestions and modifications etc.

However, because the practice of my own projects can be regarded more as curatorial practice rather than artistic practice I have also been informed by the practice-led curatorial methods of Joasia Krysa (2008) and Ele Carpenter (2008), where each examined one of their own practical projects. Due to the collaborative and participative aspects of my work, I have developed projects that have involved the production and delivery of work that explores similar processes to my own that could then be assimilated into my practice, as is the case with the work of Kate Rich and Kayle Brandon, described in James Flint's (2006) article for the Guardian, and Julian Priest, whose practice is described by the Banff Centre (2011). This community-based approach relates directly to a point raised by Stephen Kemmis and Robin McTaggart when writing on the subject of participatory research in qualitative research.

> "Three particular attributes are often used to distinguish participatory research from conventional research: shared ownership of research projects, community-based analysis of social problems, and an orientation towards community action." (2005, p.560).

This orientation towards communities of participants is necessary to my research, and is also reflected in the individual methods chosen for obtaining information for the contextual review. The contextual review includes some historical examination of the origins of open source as a defined process for project development, in order to examine what I believe has been an inaccurate use of the term 'open source' amongst some new media art projects. The contextual review also examines the PhD dissertations of others who have explored related territories, such as such as Chun Lee (2008), Joasia Krysa (2008) and Ele Carpenter (2008). Because the field is a relatively new one, with much current debate but a limited amount of literature, as well as published sources. Following an action research methodology, there is a degree of co-operative enquiry as can be seen through a chronlogical progression in my research, I began by hosting discussions on the special theme of open source on the CRUMB discussion list (see Appendix C). I selected guest respondents and led a debate on the subject of open source. This debate was very productive with over one hundred responses and many further avenues to be researched as respondents became active in this exploration of the subject. This was then followed by a research trip to Eyebeam Art and Technology Center (see Appendix E.3) where I conducted interviews with key practitioners (see Appendix B) which inform the contextual review. The interviews were informed by points raised in the earlier discussion list and aimed to obtain current opinions from a range of practitioners and in particular I aimed to get details of the process of production, which is not often described in published literature. The interviews also aimed to inform definitions of different types of open source practice and to help identify the types of hierarchies involved. This then generated further online discussion relating to residencies and labs in relation to open source, as can be seen in the *New-Media-Curating* discussion list in June 2008 (hosted by Sarah Cook). This cycle of enquiry led to a body of knowledge that informed my practice-led research.

This combination of methods aims to apply firm definitions of open source practice to my own work in order to gain a full understanding of the application of open source procedure as a method of practical project development. By analysing and reflecting upon practice, the research aims to be of use to other practitioners.

## *1.4 Scope*

The area of research covers the overlap between open source production methods and the methods of a participative art system. It is beyond the scope of this research to cover the large amount of pure media, political, commercial or social theory concerning software, and it is not a full historical analysis of new media art or open source development. Instead, it refers to commercial software and art but only in the light of a comparison and primarily to English-speaking practice in the field, within the last ten years. The participative new media art referred to is primarily visual art or mixed media.

There has been a lot of recent literature concerning participative art systems. Although it is beyond the scope of this thesis to cover this in depth, a brief outline here can summarise some key points which are relevant to the actual scope of the research. The book Rethinking Curating (Graham and Cook, 2010) gives brief overviews of some of the key debates in relation to new media art, firstly by discussing new media art in relation to space and materiality with reference to ideas of Rosalind Krauss' “post media” and the importance of processes and behaviours. Graham and Cook refer to histories of conceptual art in discussing the movement from art “object” to more immaterial processes. The chapter on participatory systems discusses ideas of Nicolas Bourriaud's “relational aesthetics” and also Claire Bishop's objections to this on the grounds of the need for space for conflict within any participatory system. Claire Bishop's edited book *Participation* (2006) includes chapters by Joseph Bueys, which make links to histories of “social sculpture” in order to understand different kinds of participation in relation to a tradition art world understanding of the artist as author.  Bishop acknowledges the limited amount of attention to pedagogy within her choices.

Others have understood issues of participation in relation to art as an “educational turn”, such as Dave Beech when describing the “turn to pedagogy within the immediate context of other artists turning to cuisine, clubbing, sport, business, therapy, leisure, spectacle, retail and communication in the pursuit of an 'art encounter'.” Beech (2010 p48). Beech also discusses participation in relation to an artworld hungry for “new publics” Beech (2007).

Graham and Cook suggest the use of Paul Baran’s network diagrams as a way to map different kinds of participation, including the “distibuted” model which is able to deal with ideas of joint authorship such as used in open source (see page 92 of this thesis).

Because these debates are reasonably well covered by the research of others, this research therefore has a more limited scope, informed by these debates. This research covers open source methods in comparison to those found in new media art through an analysis of my own artistic practice. A large part of this enquiry has been practice-led and informed by a cycle of enquiry as described earlier, beginning with a debate via the CRUMB discussion list, followed by interviews that informed further debate and practice. As such, large parts of the data will be experiential through direct engagement with groups of artists and participants. Projects take the part of fieldwork resulting from critical engagement with the subject online and at various sites and galleries located within the UK and internationally. This is in order to sample from a wide range of participants of varying backgrounds. The research also sits within the wider artistic context about how artists collaborate in de-materialised/networked environments

As Beryl Graham states when referring to the work of artist Ashok Sukumaran in her essay:  What kind of participative system? Critical vocabularies from new media art

> “The artwork is not an object, but a process. Art has, of course, been dealing with issues of immateriality, virtuality, mechanical reproduction, conceptualism, time, and audience participation for some time now, but the notable thing about new media art is that these factors have been inherent starting points for the work, from which further critical distinctions have been made.” (2010, p.283)

## *1.5 Definitions*

In this thesis, the following terms are used with the working definitions:

**Open source** in relation to software concerns open access to the precompiled source code. Programmers with access to this can then modify the code, and change the way that the software works ("to open source").

**Free** software is available without cost to the user. Open source software is not always free, but is often associated as a feature or ethic. **FLOSS** stands for Free Libre Open Source Software.

**Open source production** is, therefore, a production process that uses code that is open source. However, the leader in the Free Software/Open Source community, Bruce Perens when discussing the open source definition states that "Open Source doesn't just mean access to source code" (1998). He goes on to define a list of criteria which software must comply with in order to be distributed as open source . These definitions are a list of terms that must be accepted in order to distribute code as open source. He writes,

> "The open source definition is not in itself a software license. It is a specification of what is permissible in a software license for that software to be referred to as Open Source. These specifications cover: free redistribution, source code, derived works, integrity of authors source code, no discrimination against persons or groups, no discrimination against fields of endeavour, distribution of license, license must not be specific to a product, license must not restrict other software and license must be technology neutral." (Perens, 1998)

**Heuristic:** I use this term to highlight the process of problem solving through trial and error in both open source production process and in the collaborative and participatory practices of artists groups working in the field of new media art (a heuristic technique).

**Participative and collaborative new media artwork:** I will be exploring the rationale behind this criteria as it relates to open source via my practice.

In this thesis I use the terms **participative** and **collaborative** to cover general practice amongst contemporary practitioners who share ownership of intellectual property and distribute their working practice amongst other artists and audience members. To define these two terms in greater detail as they relate to my practice, please see below:

**Participation** –

Participation is defined by Graham and Cook in the CRUMB book Rethinking Curating in the following terms:

> *"Participation: "to have a share in or take part in."* Participation implies

> that the participant can have some kind of input that is recorded. In common language, "more interactive" can actually mean "participative" that is, not just getting reactions, but also changing the artwork's content." Graham and Cook (2010, p.113)

Participation is therefore used in this thesis to mean that the participant exerts an influence over the content of the artwork. It is fitting to refer to the concept of social sculpture as created by Joseph Beuys in order to further illustrate this point. Looking at Beuys' statement on social sculpture "I am searching for field character" from 1973, we see him raising the concept of self determination and participation in the cultural sphere:

> "EVERY HUMAN BEING IS AN ARTIST who - from his state of freedom - the position of freedom that he experiences at first hand - learns to determine the other positions in the TOTAL ARTWORK OF THE FUTURE SOCIAL ORDER. Self-determination and participation in the cultural sphere (freedom)" (Beuys 2006, p.25)

When considering the participative nature of my practice-based research projects such as the *Random Information Exchange,* it is useful to relate this definition of participation back to Beuys' installation Bureau for Direct Democracy which he set up at Documenta 5 in 1972. This structured encounters between the artist and participants (Beuys and Schwartze, 2006)

**Collaboration –**

Collaboration is defined again by Graham and Cook in the CRUMB book *Rethinking Curating* in the following terms:

> *"Collaboration: "working jointly with."* Unlike *interaction* and *participation*, the term *collaboration* implies the production of something with a degree of equality between the participants. Collaboration between people is an inherently participative and interactive process, hence the confusion between the discourses. However, collaboration in the present context is the "odd one out" because whereas interaction and participation concern primarily the relationship between the artwork and audience, collaboration usually concerns production, which may be between artists or between curators or a combination of both." (2010, p.114)

In my practice, collaboration took place between artists with similar aims all working together in the development and execution of projects. In his article Models of Authorship in New Media, Lev Manovich describes collaboration as it widely applies to new media culture.

> "The most often discussed new type of authorship associated with new media is collaboration (over the network or in person, in real time or not) between a group of artists to create a new media work / performance / event / "project." Often, no tangible objects or an even definite event like a performance ever comes out from these collaborations, but this does not matter. People meet people with common interests and start a

"project" or a series of "projects." (2002)

**New media art –**

In his essay *New Media, Art and Science* Geert Lovink describes new media art in the following, "New media arts can best be described as a transitional, hybrid art form, a multi-disciplinary "cloud" of micro-practices. Historically "new media" arose when the boundaries between clearly separated artforms such as film, theatre and photography began to blur, due to rise of digital technologies." (2005).

Lev Manovich takes a slightly differing historical view of new media art in his essay *New Media from Borges to HTML*. Arguing that the growth of new media after World War II created parallels that changed the relationship between art and technology, stating, "In the last few decades of the twentieth century, modern computing and network technology materialized certain key projects of modern art developed approximately at the same time. In the process of this materialization, the technologies overtake art." Manovich (2003).

Graham and Cook (2010) define new media art as art which behaves according to Steve Dietz's three categories of connectivity, computability and interactivity. Definitions of new media can change, I have therefore highlighted these three flexible and usable working defintions as relating to work involving software and participation.

## *1.6 Structure*

The thesis is organised with the contextual review covered in chapters 2 and 3 and the practical work covered in chapters 4, 5, 6, and 7. The interpretation of results is in chapter 8 and the conclusions are outlined in chapter 9.

A chronological narrative of the research process can be described as beginning with a historical review of early networks, which contextualises hacker culture and its part in the origins of open source culture and software development strategies. Following this review the adoption of open source methods by artists and artists' groups is examined and related back to their point of origin with hackers in the code/software development environment. This body of knowledge developed during the first year of my research then enabled me to extend my research by developing a number of practice-based projects and case studies over the mid and later stages of my research with each project examining aspects of my earlier historical enquiries.

The practical aspect of my research began by examining and further developing my work as part of the artists' group The Polytechnic, which began prior to this research period. This enabled me to examine the observable production processes and open source strategies adopted by the artists' group. This examination and continued interaction formed a solid basis for the development and execution of a number of experimental practice-led case studies that explore the different facets of open source as applied to an artistic endeavour.

Firstly, the *Datarama* project is further developed to examine how communities of developers grow around open source projects and what role the lead developer/project instigator plays in this growth. Running along side *Datarama* I further developed the *Random Information Exchange* project in order to examine the role iteration has in open source projects in cultural settings. I also used this project to gain insight into the possibility that I could minimise my personal ownership of the concept in order to observe participants' interaction with the system, exploring social hierarchy within this environment.

Building upon this growing body of knowledge and ideas generated during practical implementation of open source style projects I developed a final project that enabled me to gain insight into the roles that artists take within a development community, in order to better understand the motivating factors found amongst artists who engage in open source style projects as well as gain a cleared understanding of the benefits and limitations offered by engagement with open source principles.

# Chapter 2: Context – Early networks, technology, and software production

## *2.1 Introduction*

This chapter is a contextual review of early networks, technology and software production; it includes a brief overview of early hacker culture and how the development of computer technology influenced this culture. It examines the way in which the systems and ethics developed by the early hackers were then adopted by advocates of open source. It identifies the production processes, features and practices of these early networks of software production. The section that examines production methods compared to open source methods from this contextual review compares strategies and identifies the main features of open source software development. *Section 2.3* identifies from the contextual review some recurring common features and values shared by open source software production and participative systems in new media art. *Section 2.4* identifies from the contextual review approaches to collaboration which do not impose a preventative ethical or a political stance upon the open source framework.

## *2.2 Background – Software production methods*

### 2.2.1 Historical background

There is a substantial body of writing about value systems relating to technology, networks and software, including work by Steven Levy (1984), Eric S. Raymond (1999), Tiziana Terranova (2004), Gabriella Coleman (2004), Charles Leadbeater (2008). It is beyond the scope of this thesis to cover a full history in detail, but it is useful to consider briefly some of the ways in which older methods feed into current goals and practices.

The development of computer technology has created forms of collaborative behaviour that have carried through into current practice. From early on in the field of computer science, time in the machine room had to be organised. This was so that programmers had a time-shared slot at the computer that was large, very expensive and high maintenance. Social hierarchy grew around who had “Priority status for running jobs in the machine room” (Weinberg 1998, p.11). One of the ways to become a member of the hacker elite at places like MIT (Massachusetts Institute of Technology) was to take someone else’s program and improve on it (Turner 2006, p.134). For this situation to be possible it was important that hackers made their code openly available to one another, thus making it open to modification. Academics and hackers began to use networked computers as a means of exchanging information. As technology advanced and computing became more accessible to hobbyists followed by general users, the network protocols also became more accessible. Barbrook (2007, p.277) states, “Quite spontaneously, non-academics began to adopt the Net’s academic ways of working.” I would agree that the academic methods became commonplace, however, I would not call this situation spontaneous. Instead, I believe that it is inherent behaviour that must be adopted in order to interact successfully with computer technology and communicate this process with other users of computer technology. Before the Internet became commonplace, hackers, academics and hobbyists would use a bulletin board system as a means of collaborative communication.

Bulletin boards are a good example of early systems for collaboration. An electronic bulletin board system or BBS in its simplest form consisted of a computer running in a house or garage that other users could connect to via a modem and a phone line. Once connected to the system, they could read and post messages. In theory, this system was open to anyone who had access to a computer. A good example of the follow through of hacking culture into a wider domain is The WELL (Whole Earth ’Lectronic Link) this was set up by Stewart Brand who had previously set up The Whole Earth Catalog (1986). Another example of a BBS that adopted the early hacker tendency towards openness was CommuniTree.

> “Within a few months of the first BBS’s appearance, a San Francisco group headed by John James, a programmer and visionary thinker, had developed the idea that the BBS was a virtual community, a community that promised radical transformation of existing society and the emergence of new social forms.” (Stone 1991, p.88)

Going into the 80s, it became possible for wider groups of people to interact online. Terranova describes online exchanges as “experimentation with the peculiar semi-fluid mechanics of network space” (Terranova 2004, p.69). She then goes on to describe CommuniTree as an example of “failed experiments in network hydrodynamics, vortexes that dissolved under the tides of the network”. CommuniTree was destroyed by hordes of teenagers with access to Apple II computers in their school classrooms when they filled up the Board’s 300k of disk space with expletives and disruptive text.

CommuniTree had no gatekeepers and this event lead to the introduction of control and surveillance tools in future BBSs. The BBS became a place of cross genesis between hackers, activists and artists. One such place where this happened was a BBS called The Thing. In an interview with one of the founding members of The Thing, G.H. Hovagimyan (see Appendix B.3.), it became apparent that at this early stage in the development of digital networks the hacker ethic had carried through. G.H. traced the development of the early bulletin boards through to Internet forums and demonstrated how he managed to use this system to bypass the standard filters inherent in the

New York arts scene at the time.

> “I had done a piece called *Barbie and Ken, Politically Correct*. So I scanned them into GIFs and put up one a week for 3 months as a file attachment. If you were a member of the BBS you could download them... I was taking my work around all these galleries. Soho was the arts scene and there was one gallery called TZ art. So I was there with my hat in my hand with my slides etc. and it wasn't going anywhere. So then he got called away to something and he went out of his office and I looked on his computer and BKPC was playing on his computer as a screen saver, they were just coming up one after the other and I said ‘Ahh’. It was totally perfect because my work was in this gallery.” (G.H. Hovagimyan, interview, Appendix B.3.)

This illustrates the BBS’s potential to bypass the traditional system.

It is important to differentiate between what the BBS was at this stage and what would later be known as bulletin boards or Internet forums. The early BBSs mirrored the nature of physical bulletin boards, i.e. one person posting information for many to read. Subsequent Internet bulletin boards have become places for mass communication in a way that requires far less understanding of computer systems and networks protocols. Allucquère Rosanne Stone writes,

> “The CommuniTree Group proposed a new kind of BBS that they called a tree-structured conference, employing as a working metaphor both the binary tree protocols in computer science and also the organic qualities of trees as such appropriate to the 1970s. Each branch of the tree was to be a separate conference that grew naturally out of its root message by virtue of each subsequent message that was attached to it. Conferences that lacked participation would cease to grow, but would remain on-line as archives of failed discourse and as potential sources of inspiration for other, more flourishing conferences.” (Stone 1991, p.89)

The system adopted by CommuniTree is again an illustration of the behaviour necessary for networked discourse. The tree-structured BBS appears to be an idealistic device but in many ways it is a chance encounter between a group of individuals who behave in a certain way and a system that allows this behaviour to flourish.

A typical BBS would have a sysop (systems operator). This would generally be the person who had a computer in their house running the BBS software and they would perform standard tasks such as updating the software, backing up and maintaining the hardware etc. However, they would also be in a position to regulate content alongside other appointed co-sysops, thus placing them in the position of gatekeeper where they could moderate the flow of information in order to maintain quality control and avoid events similar to those that led to the demise of CommuniTree.

It became apparent that a gatekeeper was necessary in order to maintain quality control. This can be witnessed in many of today's systems for gathering and exchanging data. Wikipedia is a good example of a current system that employs the gatekeeper system in order to ensure information is of a high standard. It is apparent that gatekeepers are motivated to attain a level of kudos within their own sphere of interest, although there is also a genuine altruism and in most cases it is mix of the two. A good example of a Wikipedia gatekeeper is Orange Mike who states on his user page,  "I am considered to be toward the 'deletionist' end of the inclusionist-deletionist spectrum, because I believe that many articles here (including not only new ones but many existing ones) are violations of our standards" (Lowrey 2010).

The Wikipedia/Orange Mike style gatekeeper is in contrast to how John Lilly, Chief Executive Officer, Mozilla Firefox, sees things working at Mozilla. Mozilla (2011) describes its mission as being "to promote openness, innovation and opportunity on the web". In an interview with Peter Carey (2008) of San Jose Mercury News, John Lilly describes Mozilla as a "chaordic [sic]" organisation explaining:

> "The idea is that you try to take responsibility and authority and decision-making and push it as far to the edges as possible. Lots of systems work that way, like the Internet itself. They tend to result in systems that are unpredictable at some level but also very strong, very robust, very tolerant of people coming and going and contributing, and tend to result in unexpected innovation." (Carey 2008)

This contrast could exist due to the different nature of the two products. Wikipedia is a place on the net where you can get information. Mozilla make tools to access the web with a neutral opinion about what you should do once you are online. Nevertheless, gatekeepers still exist amongst communities of programmers and software engineers. Mozilla has also produced a browser that provides a genuine open source alternative to the prevalent Internet explorer.

Moving on to more contemporary literature concerning networks and open source, many publications concerning this area take a broad starting point of describing the wider culture of computer technology in relation to today's society and politics, Lovink (2002); Lyotard (1993). Eric S. Raymond (1999) has more specifically discussed open source and Linux in relation to systems of work, production and distribution, and has described and compared "the cathedral and the bazaar" as analogies of the ways in which these systems work. Raymond further defined and rationalised a number of terms and behaviours associated with open source in the collection of essays published as *The Cathedral and the Bazaar*. (1999)  In *Just For Fun: The Story of an Accidental Revolutionary*, the autobiography of Linus Torvalds, the software engineer who initiated the development of the Linux operating system, there are many references to undefined hierarchical structures within the open source community (Torvalds and Diamond 2002). More recent technical books such as Barnes (2008) provide useful technical information, and more recently still, useful publications have emerged which cross over from being purely technical into discussing the creative merits of computer programming such as *Visualizing Data*, (Fry 2008), And *Processing,* (Reas and Fry 2007). These publications provide a body of work that examines and defines computer and network culture in relation to open source and its impact on software developers before moving on to creative practitioners with Fry and Reas. This knowledge enables us to gain a clearer picture of emerging themes and definitions.

### 2.2.2 Software production methods compared to open source methods

In moving towards working definitions of open source methods, a comparison of the way in which the two groups use the same production methods is useful. Many programming methods and procedures used by developers working within an open source project can mistakenly be seen as open source methods. There is a great deal of crossover in the choice of procedures followed by both commercial software developers and open source software developers. In fact, many commercial companies also contribute towards open source software. The key differences are the systems for allowing public contributions and the choice of license attached to the project, which then allows for free use. The following table compares commercial and open source use of similar strategies:

| | | **Commercial software production** | **Open source software production** |
|---|---|---|---|
| **Initial Idea** | | Consumer demand | Felt need |
| **Development/ Licensing:** | **Iterative** | To begin project while discerning client's needs.<br>For efficiency reasons. | Communal development in repeated cycles. |
| | **Modular** | Able to adapt to commercial demand. | Reuse of parts and non-parallel working. |
| | **Versioning** | Management of large code base in private. | Management of large code base in public. |
| **Code repository** | | Private | Public |
| **Release cycle** | | Alpha and Beta release stay in house. Public release of Gold version. | Wide availability of Beta. Often never leaving Beta state. |
| **Distribution** | | Commercial | Open Source |

*Table 1: Commercial software production processes compared to open*

*source.*

The software production method can be understood by comparing the following common strategies that are used both in commercial and open source software development:

**Idea**

In commercial software development the initial idea for a software project can arise from consumer demand whereas in many cases an open source project begins out of frustration, i.e. someone needs a tool to complete a specific task that either does not exist with enough freedoms or is only available at a prohibitive cost.

**Development/licensing**

Licensing sets out a commercial framework for interaction and how people share in a project. It allows for the free reuse of existing parts of other software. Licensing manages non-parallel working methods if managed properly by a gatekeeper.

Commercial software is vigorously protected through a number of legal means in order for the development company to retain full ownership of their intellectual property. When purchasing commercial software you are, in fact, purchasing a license to use a copy of the software with little or no input into how it is developed. Open source provides a way to protect collaborative work via a license. If this protection is not in place, then the software and code base it is at best a piece of public domain work that is exposed to plagiarism and is at greater risk of forking, thus dividing its workforce. Licenses also provide a framework for engagement with a project’s code and intellectual property giving clear guidance to those interested in modifying existing work.

Work should be able to be copied and easily distributed from multiple points. This is made possible because of the Internet and what Terranova (2004, p74) describes as “disintermediation (embodying the famous death of the middleman from bookshops to travel agencies and computer stores), but also the means through which a flexible, collective network intelligence has come into being.” As was mentioned earlier, with commercial software you are purchasing a license to use a copy; with open source the license accompanies the software and can be freely distributed.

**Iterative**

This is a situation where a project develops incrementally, gradually acquiring complexity as the needs of the software become clear. This is a model that has been favoured for years by proprietary software developers as this allows them to gradually define a project with a client who does not know what they want (which is actually very common). For example, a client contacts the developers asking for software to help them keep track of stock in their warehouse. The developers can begin developing a simple database structure and graphical interface whilst working with the client to define their overall needs. Complexity is achieved through repeated development cycles, these cycles are very often present in open source software development in the form of code sprints.

**Modularity**
This is the separation and combination of elements as discussed by Lev Manovich (2001, p.31). Again, there is a common belief that this originated as an open source development method due to the benefits of a modular approach to programming as witnessed within a freely distributable environment. Open Frameworks is a good example of this. At its core, Open Frameworks is a set of programming libraries that make it possible for artists and interactive designers to create tools in an open development environment. It ensures commercial software is able to adapt to new demands and avoids duplication of effort in open source developments.

**Versioning**
Versioning systems such as CVS (Concurrent Versioning Systems) are central to large-scale distributed open source projects. This system is an automated code management system allowing programmers to upload changes to the code base that can be logged and checked in without requiring the re-upload of the entire code base.

**Code repository**
A software or code repository is the central point at which a software project's intellectual property is stored and managed. It is often linked with a versioning system. However, it is worth separating out in order to highlight the main difference between the commercial and open source use of a code repository. The commercial project restricts access to the code repository to a limited number of developers, but the repository for an open source project is open to the public to access.

**The release cycle**
The software release cycle is often viewed as an open source software development method. This is the distribution of a software project as it goes through its development stages beginning with pre Alpha through Beta until there is a release candidate, followed, if successful, by a Gold release to manufacturing/marketing. It is only very recently that commercial software developers have began publicly releasing software in its Beta state. This is mostly down to the success of this system within open source projects that are often in a perpetual state of public Beta release.

**Distribution**
Distribution for commercial software is paying for the license to use it. With open source projects you are free to distribute the software and code, along with the license. It can be done by anyone with the ability to copy the software and it is non-centralised.

## *2.3 Features of early networks*

### 2.3.1 Freedom (politics and economics)

A well known software developer and software freedom activist Richard Stallman, whose biography can be found on Wikipedia (2011d) was working at MIT when they installed a password system in 1977. According to Levy (1984, p.417) Stallman, who was working in the MIT Artificial Intelligence Lab at the time, emailed his colleagues telling them that he had made his password blank and that they should do the same. This was both in order to maintain a previously open system that had allowed them to work quickly and effectively and a way of refusing to play along with a non-hacker ethic by liberating his password.

This leads us to a discussion in which Armin Medosch (2006, p.238) describes the open embedded system in his essay *Meshing in the Future – The free configuration of everybody and everything with Hive Networks*. The open embedded Linux system is a way of replacing the pre-loaded firmware (embedded software) on a device with Linux, thus providing the possibility of using the hardware to its full potential. Manufacturers disable many features in order to sell them later as a new feature, therefore, what Stallman did with the MIT password hack was to say that hackers have the right to freedom, i.e. they refuse to be limited by the needs of the existing models followed by their employers, and they have the right to use their hardware and systems to their full potential.

The use of copyright free material enables open source software developers to defend the freedoms they require in order to improve existing technologies for their own uses. This also means that open source software must allow for programmers and hackers to make modifications that can then be shared and improved upon by embedding the freedom to duplicate and distribute into the accompanying license.

### 2.3.2 Sharing

A history of hacker culture dating back to time-share machines can be linked to a desire for a system that reclaims freedom after aggressive proprietary software trends have locked access to intellectual property.

Hackers recognised very early on that innovation came through the open sharing of ideas as a result of working in close quarters at a time when computer time was a valuable commodity. Computer time became regarded as a status symbol with people higher up the pecking order being generally the least able to use it constructively (the higher up the management scale, the more outdated the programming skills). This ethic still prevails in hacking circles today, with more kudos being awarded to the hacker writing good code on an old Linux laptop than would be awarded to the same on an expensive Apple Mac.

In order for sharing to be possible, a number of project resources need to be freely available in a form that can be distributed across a computer network.

Software code must exist in a central location so that any changes can be tracked and peer reviewed. The technology and materials used (if the project involves hardware hacking) must be available to the majority of users. An exchange of knowledge, method and ideas must also take place. This exchange occurs vial mailing lists, IRC and Internet forums, all of which reflect early BBS practice.

### 2.3.3 Recognition of skills

Recognition of skills amongst peers can also aid in the progression through a project hierarchy into a gatekeeping role. This is a recognised status that can bring many benefits, such as the opportunity to display your work to a wider audience.

Recognition was awarded to social protagonists such as Perens (1999) for his work on the open source definitions (as acknowledged in the book *Open Sources* where he presents the definitions with commentary) and Stallman in the form of honorary doctorates and awards listed in the recognition section of the Wikipedia (2011d) page about him. Raymond (2010) became a recognised spokesperson through his many articles about open source and hacker culture. Torvalds has been recognised with awards from academia, industry and the media as are listen on Wikipedia (2011e). Torvalds has even had a meteorite named after him, as described by NASA (2010). There are many other great coders yet they remain mostly anonymous, although we know the names of the community leaders. Recognition goes to the 'wise' leaders who take advantage of informal structures; structures that have been influenced by the nature of computing.

Raymond (1999b, p82) discusses craftsmanship and this is a selling point for the open source initiative when encouraging corporate take up. Open source methods of development allow access to peers and support good craftsmanship. This peer group can provide feedback and kudos: the exchange of code over computer networks makes this possible as kudos is awarded for innovative code and improvement of existing code. The example of G.H. in section 2.2.1 creating work for The Thing shows how this kind of peer review can have a beneficial effect and can, in fact, encourage take up from new participants.

### 2.3.4 Summary of features of early networks

A distinguishing feature of early hacker networks is the recognition that the computer network allowed for a decentralised framework. Thus, someone working at Bell Labs and someone working at MIT would be developing a shared ethic that went beyond local frameworks. Hackers realised this practice flourished when it was not restricted and so defended the freedom they needed to work collaboratively which then had an influence upon the development of open source. Sharing of resources originates from the management of time-shared computers. The benefits of sharing resources have continued through into open source practice with the freedom required to do this embedded in the system. Recognition of skills through peer review required access to computer networks and freedom in order to share, modify and improve code.

## *2.4 Early networks in practice*

Some features of early networks in practice include the following.

### 2.4.1 Agnostic system

When addressing free open source software developers' political agnosticism, Gabriella Coleman (2004, p509) states:

> "To understand the logic of political denial, it is instructive to define the rationale for freedom formulated in the philosophical underpinnings of FOSS licenses. The moral and semiotic load of free software is a commitment to prevent limiting the freedom of others."

Open source would definitely limit its potential pool of developers if it were to take an overtly political stance, and the Open Source Definition by Perens (1999) seems to place more importance on the strength of this pool than upon any political or ethical position which makes the Open Source Definition appear attractive to corporate interest.

The Free Software Definition as maintained by the Free Software Foundation (2010), on the other hand, maintains that Freedom Zero is the freedom to run any program for any purpose. They then go on to clarify:

> "It is the user's purpose that matters, not the developer's purpose; you as a user are free to run the program for your purposes, and if you distribute it to someone else, she is then free to run it for her purposes, but you are not entitled to impose your purposes on her."

In The Open Source Definition (Annotated) as it appears on The Open Source Initiative  website (n.d.) , we get to point five before we reach a similar aim:

> "No discrimination against persons or groups – the license must not discriminate against any person or group of persons…" with the rationale: "In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process."

There is a difference between these two aims. With the Free Software Definition, all definitions following zero cannot work unless point zero is upheld. With point five of the Open Source Definition, it would seem that it is upheld in order for the system to get maximum benefit from as large a number of developers as possible. It is also the case that when choosing an appropriate Creative Commons license, there are no options for excluding users on ethical grounds.

### 2.4.2 Awareness of other developers' ethics

The original hacker ethic stems from use of time-share computers as discussed by Levy (1984, p39):

> "The precepts of this revolutionary Hacker Ethic were not so much debated and discussed as silently agreed upon. No manifestos were issued. No missionaries tried to gather converts. The computer did the converting."

### 2.4.3 Collaborative behaviour

An example of collaborative behaviour is The Whole Earth Catalog, first published 1968 as a "single textual space" according to Turner (2006, p72). Turner calls the Whole Earth Catalog a "network forum – a space where members of these communities came together, exchanged ideas and legitimacy, and in the process synthesized new intellectual frameworks and new social networks." (2006, p72) Turner also states that "the catalog itself, helped create the cultural conditions under which microcomputers and computer networks could be imagined as tools of liberation."(2006, p74).

Steve Jobs in a Commencement address at Stanford University described the Whole Earth Catalog as "made with typewriters, scissors, and Polaroid cameras. It was sort of like Google in paperback form, 35 years before Google came along: it was idealistic, and overflowing with neat tools and great notions." (Jobs, 2005)

### 2.4.4 Hive mentality

From simple systems we get simple heuristics that allow for the development of complexity and sophistication.

Programming groups always find a way to structure their work. Gerald M. Weinberg when talking about organisational structure said:

> "an informal structure always grows to correct and complement the work of whatever formal structure exists. Sometimes if the powers-that-be are sufficiently wise, innovations in the informal structure can be implemented formally, although not always as exact equivalents." Weinberg (1998, p48)

Open source is an extension of this self-structuring, with certain leaders taking the role of the wise in order to formalise the procedures

### 2.4.5 Summary of early networks in practice

There is a traceable route from the ideals of the 60s counterculture through its influence on computer scientists and academics. This provided structure and influenced the ethics that grew from time-share computing and went on to influence development of the UNIX operating system and the mind-set of its users.

## *2.5 Summary of Chapter 2: Early networks*

Hackers developed collaborative behaviour due to the time and resource pressured nature of the machine room. This behaviour spread to early computer hobbyists as network resources were limited and costly. The bulletin board system helped spread the hacker tendency towards openness whilst further developing a system for communication and collaboration influenced by electronic network protocols.

Successful projects have gatekeepers who maintain a project's quality and ensure that information flows. The behaviour of a gatekeeper is governed according to the project goals.

There is crossover between open source software development and proprietary software development. Many computer programmers contribute to both. The software production method can be broken down into a series of strategies. There are, however, key strategies that identify a project as open source

The freedom inherent in open source software development makes it attractive to developers and users who already have strong views with regard to aspects of ethical and political freedom and personal liberty. Others engage with the freedom inherent in open source software as a result of the desire to improve their craftsmanship, achieve recognition of their skills and participate in an innovative process. These goals are easier to achieve when engaging with code that is freely available and open to free use and modification.

Early hacker culture showed that there was a desire to explore a system's potential beyond the limited needs of their employers. The adoption of this system amongst artists, as I will discuss in the following chapter, also reflects this desire. Open source methods can be applied to the creation of new media art and provide artists with a framework to create work outside of the established system.

# Appendix B Interviews

### *B.4. Interview with Michael Mandiberg*

The interview with Michael Mandiberg was conducted in New York on the23rd of October 2008. Michael Mandiberg was then a fellow at Eyebeam Art and Technology Center. An edited version of this interview was later published on the CRUMB website www.crumbweb.org

Interviewer: Dominic Smith
Interviewee: Michael Mandiberg
Dominic Smith: I was hoping to have a chat with you about your experiences of open source, your initial experiences, your current experience and where you stand on it.

Michael Mandiberg: I learned to code by looking at source code on websites. My initial experience with any kind of programming was by default open source. I was doing graphics before so I've been using Photoshop since version 2.5.1. But when talking about code I learned to code largely by looking at other people's code. I am going to give a presentation at Berkeley in two weeks at the Takeovers and Makeovers conference which is about intellectual property, artists and appropriation.

I'm going to frame my talk through my realization that I was making Open Source work without realising it. I did this project called AfterSherrieLevine.com. Sherrie Levine photographed Walker Evans' work and called it her own work. The funny thing is, it was a form of Commons enclosure. Although I didn't have the words "Commons enclosure" at the time to describe it, these are the words I would use now. She had a legal case with the Walker Evans estate and ended up shutting things down. I wouldn't have called it open source at the time, but that's what I did when I scanned the works out of the same edition of the book and put them up online as hi-res images and certificates of authenticity which you print and sign yourself.

So there is 'open source' the noun, and then there are two different versions of the verb 'open source', 'to open source'. So you're working on a project and you release it open source, that's to open source a project. But the other version of open source is a certain kind of reverse engineering, it's kind of hostile or confrontational, and it's to open source somebody else. I was open sourcing Sherrie Levine in a sense. So I think that a lot of my work comes from that appropriation and that's a starting point. I did a bunch of programming, a series of Firefox plug-ins that are open source, again they are open source largely through the fact that they are written in JavaScript.

During the last year I have been doing a lot of design-related projects. I've been working on a reflective bicycle and on lampshades. I have found it really interesting because everyone wants to buy these lampshades, but no one wants to make them. I made an instructable but no one has really done it, people want to buy them, people want the easy way out. We released the instructable with a Creative Commons Attribution (CC-BY) license. Basically, all we want is to say 'Made in the Eyebeam R&D lab', that's all we want. We made a video and blog posts, and sent it out into the world. It’s got 30,000 - 40,000 views - maybe more - I haven't checked. The video explicitly says, “this is just a video to get it out into the hands of someone who will actually manufacture the lampshades, we are giving it away, make money off us.” But we are having to work very hard to give something away. It's a very strange thing; and this is part of what the talk I'm going to be giving in 2 weeks is about, it’s this notion of having to work hard to give something away. Unless you turn yourself into a business your idea stays as a sort of prototype. Cutting these things [lampshade parts] out on a laser cutter is time-consuming, tedious and totally inefficient. It is great for prototyping but the lampshades need to be die cut. They need to be die cut on a scale of 10,000 at a time and the costs will drop by an order of magnitude. I don't want to do that, I don't even know how to do that. Certainly whether I want to or not, I don't have the skills, I don't know how to get into the distribution channels. But someone must - Ikea, Walmart, Target, Urban Outfitters, whatever. I realise that finding those people is really hard and there might be a little bit of trepidation like, “Do we have the rights to do this?”

I'm in the process of finishing a design textbook called Digital Foundations with Xtine Burrough. The book is a mash-up of the Bauhaus basic course and the Adobe Creative Suite. So we're talking about all these exercises and principles that form the cornerstone of visual education and design in arts education in the 20th Century to teach 21st Century tools. No one has done this, no one has even considered the way that the colour picker in Photoshop is derived entirely from Josef Albers’ colour theory exercises, and the idea that you can teach both at the same time is super powerful.

We spent (by "we" I mean me and my co-author Xtine Burrough) nine months negotiating the contract because we wanted a Creative Commons license. We got the first Creative Commons license out of Peachpit and New Riders. It is being published by AIGA Design Press. That was a huge accomplishment, it was about trying to explain to these businesses how Creative Commons is not a threat to them, in fact it’s a benefit. So we went through a whole lot of explaining, constantly reassuring and giving them case studies. I need to write a ‘How to’ because we're one of the first people to do this in an adversarial context, even the other people listed as being major Creative Commons authors in this kind of context, in fact, only have Creative Commons license on their online version, which is their manuscript.

One of the central ways we were able to convince them was by showing our partnership with FLOSS Manuals and Adam Hyde. Adam and I have been talking for six or seven months about translating the book into Gimp, Inkscape and the other key Open Source apps. There are these core ideas and it is about separating the ideas and the exercises from the button pushing. So via the Creative Commons license we were able to say, “Look, we have an immediate partner who wants to expand this idea”, and I have to spin it to them in words they understand, i.e. build a brand, win mind share then market share, all this marketing stuff, that's what they understood. We were able to do it. So we are going to start once we finish the actual, real book which I am crazily working on right now. Adam, Xtine and I are going to start figuring out how to translate the book into the Open Source applications, and then (because Floss Manuals is heavily translation based) it will then be translated into Farsi, etc.

DS: So how did the publishers react to the idea that it would be translated for free into different languages? Did they see it as an added value?

MM: Yes, I just kept saying you are not going to get someone to buy the Farsi translation rights, and you are not going to get someone to buy the open source translation rights. You are losing nothing, and gaining much. For me the goal of this book has been to change the way design education happens at the introductory level. I teach in a classroom when I’m not here at Eyebeam; it's a computer lab, there are fifteen Macintoshes in it. I get a bunch of nineteen year old students who have never been in a drawing classroom, they have never been in a dirty arts classroom. They have never drawn, and if they have, it was in high school, but they don't have those fundamental visual skills. This is true across the country in these design departments and visual communication departments as they have to spend money on computers, and they have to have places to put them - the drawing classrooms get cleaned out, scrubbed down and turned into a computer lab. So we have lost the location in which we would teach visual skills. What we are trying to do is find a way to bring back that 'introduction to visuality' into a computer class. For me the goal in all of this is to make a change, and I have made it clear to the publisher that's our goal and that it is good for them because it builds their brand, and using the words they understand, building a brand as an educational reformer is a positive thing. When you frame it that way, getting into as many languages as possible, they like it. Realistically they do sell the rights to translate it, they will probably sell a Spanish, German, French and Italian version. If the Chinese want to copy it they are just going to copy it, and I think they would be better off by having some sort of control over it by making it part of the license that allows them to do that.

DS: That's your experience of being a leader in that area. Have you got any experience of being lower down on the food chain?

MM: Yes, I've worked a lot with Drupal. There's a number of roles one can play in an open source community. One can be a project keeper and be the head architect type person. One can be the architect of one small little area that the community is working on. One can be someone who does small little bits of coding for those groups and one can be an end user. I have been somewhere in between an end user and a maker in several contexts. I tried to put together a template system for Drupal that was focused on artists' projects, it was a great learning experience. But I was running too many projects at once and it didn't end up coming out the way I wanted it to. I think somewhere on Drupal I'm considered a maintainer of a project that is dormant (laughs). I have also submitted bug revisions and code snippets that correct bugs in current code, and all those parts of being a contributing member/user of an Open Source project. It's all part of it.

DS: In terms of your own work, do you make saleable pieces of work?

MM: I'm making the first saleable piece of work that I've made in ten years as we speak (laughs). I'm making drawings with a laser cutter that are going to be sold as part of the Rhizome Benefit. They're going to offer them as perks if you give a lot of money, $500 or $1000; I'm not sure the exact price. I'm actually interested, and becoming more interested, in physical objects. There are some laser cutter drawings that I've been working on, that are somewhat saleable. Some of them are based on a font that I made - that I obviously have released open source - that is an adaptation of another font. I took Zapfino, it's a really decorative Baroque font that has fourteen hundred ligatures, which has variations, for example if A is next to B it looks different than if A is next to C, it's super crazy. Anyhow, I removed all of the counters - the empty space in the middle of letters like “a” and “o” and “d” - so that it could be stencil-friendly. So I've been making lots of that stuff, and all the drawings I'm releasing I'm going to write instructables for them. All the files are available if you want to mess around with them.

DS: So in the development of that work various elements of it are open sourced before it comes to the point of final release. Have you any experience of anybody taking up part of that project, enhancing it and getting back to you?

MM: I've had people make modifications.

DS: And have you then taken those on board for your final work?

MM: Sometimes. There's always the situation where someone was like “Aah this code's buggy here's a patch”, so that's one way. Another time someone wants to make a variation of your plug-in and they cite it as the start of where they went. A lot of people just grab chunks of code from it if I've done some things in my plug-ins that aren't so well documented, but I've figured out ways of doing them. And I guess, to come back to that 'After Sherry Levine' example, you can find aftermichaelmandiberg.com, it's a perfect copy of aftersherrielevine.com (laughs). So there's a whole variety of things.

Right now I'm working with reflective materials for bicycles. I am working with a couple of fashion designers, a designer called Nikolai Rose, we are trying to make a pinstripe fabric and they are going to cut a suit out of it. And I've been in discussions with a designer named Ethan Benton who has been working with retro-reflective materials. It's a dialogue, we are working together; Ethan and I shared some ideas, but decided that we probably should work on our own projects, but share ideas. I'm going to be out in San Francisco and I'm going to talk to these other guys that are doing these reflective pants. With bicyclists you have to roll your pant sleeve up, well, the inside of the bottom of the pant is covered in reflective material, so this part becomes super reflective, and they even have these little things that you slip into your pocket which you pull out, so you get little flaps that are reflective. So there is a lot of information sharing that I think is also really important, it's less hierarchical, less like, “I'm going to contribute to your project”, or, “I took your project and I made something with it”. It’s more two ways and less one way.

DS: Do you think that would have happened if open source didn't exist, or do you think that there have been other structures already invented that encourage people to work in a more collaborative fashion?

MM: I think it is because there is a culture that encourages sharing, the culture of the Internet. It’s a reinvigoration of craft culture that is so strong right now. I think that that frames a context where people are like, “Sure, come over - let me show you what I'm doing”. I don't think that any of the people that I'm in that dialogue with, would explicitly use the word Open Source. It comes two steps removed and it has as much to do with crafting and hobbyist-ism and knitting and sewing, as it has to do with computer programming, I think all of that gets channeled through the Internet.

### *B.5. Interview with Magdalena Sawon and Tamas Banovich*

The interview with Magdalena Sawon and Tamas Banovich was conducted in New York in October 2008. Sawon and Banovich are co-directors of Postmasters Gallery.

DS = Dominic Smith
MS = Magdalena Sawon
TB = Tamas Banovich

TB: My kind of observation, what happened especially what happened here. What happened with artists working with technology and digital but basically it's a generalisation but this is what I feel, basically people that are into innovation and entrepreneurship, those people kind of over run this whole field. It used to be for years that art is innovative and the innovation went into the mainstream when people innovated new uses for existing technology. Right now I have the impression that there are so many smart people, so many innovators and so many entrepreneurial people who have basically run all over this field. They just come up faster and are more involved and more engaged and they try and make an enterprise out of doing new things. I think they are just as creative and covering all the angles that artists used to. That's just my general theory. A couple of years ago I started a mobile festival, going to bring general creativity to more aspects of mobile communications to cross the so called digital divide and it's global and much better than the internet in a way. But I realised that basically all those ideas are already incredibly well covered by society and just what's going on. I was very interested to create something or to find people who create something useful, not only people in the US and a couple of very advanced countries but for all those hundreds of millions of people who are using a platform were there isn't even any electricity. You know you can get any instructions you can imagine, how to make shoes etc. So there is very little. Of course there is always an angle Artists can cover as a rule of thumb. But I discarded that, I did not continue it any further because my primary reason not to get some nice new artwork for the phone. I wanted to get something more meaningful. So that's what I feel now. There is a whole area of mobile recycling because if you go to any developing country they can take all this stuff apart, the can fix them, they can do anything with them. So there is that recycling going on, it's very efficient and it's amazingly creative. So a big question for me, what can artists anymore give to this?

DS: Yes, I really take on board what you are saying about there being too many smart people out there trying, all trying, to carve out a niche for themselves. The thing about an open source model is that all source code is freely available so if you produce something and put the code out there. If you are a charismatic person with a good idea you will be able to source people from a crowd and filter their skill levels to decide what stage in the project they work on. I am looking at this process like an open Madonna factory. In the Madonna factory if someone was good at drawing hands then they go that job, it was all divided up into sections. In a lot of respects, what I am noticing about big named artists involved in open source is that they manage to source other intelligent people with very specific skills, so someone will handle the network problems etc.

MS: Yes, but historically with art, outsourcing and fabrication is not a different issue. At certain levels of conceptual production you have an idea and you don't have the tools. So an artist is not restricted to a particular medium and we see that much more these days. I have an artist that has never touched a computer but has an idea for the piece that requires high end technical knowledge and he doesn't have it so, of course, that gets farmed out to somebody else, it doesn't make this piece any less interesting or any less meaningful. I think right now these kinds of balances between people who write their own code and are technically very proficient and you know they understand how the car runs and other people know that you push the button here and it drives.

TB: I understand were you are coming from. I remember when I was interested in digital art and there was a lot of people who had really solid programming skills. Even a couple of weeks ago I was talking to someone who said "oh I'm working with artists because I can program" and I said what kind of programming do you do and he really was just talking about being able to use high end tools in a very proficient way. It’s not really programming. Java is a programming language but that is also already a language where you use a library so you get people who can use a library, but they don't know how to make a library. That's also changed. It’s not good or bad, it’s just an interesting thing that can be used. But strictly open source? It can work but, in a way I don't see such a big difference between entrepreneurs and open source because I think open source has a big social value when it comes to big things but on the other hand I think for small things I don't see that much of a value proposition. Looking at the iPhone and all the applications, why shouldn't someone who does these things get 99cents for their little application because it is useful and then all these open source people when they grow up they live out of this, one way or another. Whether it’s being an advisor etc. So they create their own economy, it’s a great thing that open source is free but at the same time they have to live off something, so they do. You know they have family etc. So all these open source people who grow up and do very well, maybe not like Gates, but they do very well from being exceptional software engineers. So it’s an interesting thing but for me it is not so clear cut.

MS: Also on a more simple level I think art projects are not progressive in a way that, you know someone wants to do this, and then they want to do something else, so it’s not really the point that they will do something and you can give it to the next person but they are so specific and so individually geared to this or that project.

TB: I was very interested in this whole idea of a different economy but imagine that someone would do something and then another artist comes along and says “oh I can do it better” (laughs). I mean this is an interesting concept maybe I will do it (laughs). But this is just the setting, you know, he just did something and I can improve it until it’s a common free contribution to the world. Unfortunately I really believed ten years ago but it somehow digital medium can create at least an alternate economic model for art as well. But it didn't happen, all the artists who did digital art they somehow figured out how to make it unique and how to commodify it before I or anybody else could make a serious effort to create a different model because they just figured out much faster that it is maybe not a winning proposition.

DS: One of the arguments for artists using this (open source model) was put forward by Saul Abraham. To paraphrase him he said there are so many galleries, teaching posts and commissions etc but there are so many more artists and how do those artists get a foot on the ladder and gain access to peer review, gain experience, show their own work and begin to experience collaborative practice as we all do as artists. His argument is for open source as it exposes artists to a value system for their work that they couldn't otherwise gain. So how do you place that value system against the more traditional system in galleries and amongst people who have to sell work to survive.

MS: I can’t necessarily say I understand because I don't think art production and the gallery is exclusively about commerce and there is this ideal model that we share and there is this horrible idea of the gallery where we sell. Good / Bad. I think it is much more grey and much more complex, it goes different ways. We certainly don't produce shows for sale. Things sell, but you know what people show in galleries is somehow reflective of what is out there , what they see. How we started working with digital work, this impulse that it is a commodity did not enter. Yes I thought there would be enough curiosity and like with every medium at some point markets develop. It took much longer, but yes it developed, so yes there is a commodity, but it is still nowhere close to traditional media. I kind of don't know where and how to apply the idea of open source to this individual production and all these different people doing different things.

TB: If you look back, forget about digital art, just generally art and how people started to do art in art schools to being recognised artists. I mean it’s kind of an extra process because in the beginning you are sheltered from commercial system and sheltered for a period of time. So there is a period of communication and organisation and self generation groups where these people kind of open source and they exchange ideas and show each other work. It does exist, this whole system exists. Then the hypocritical part of it is the “we are better because are not selling”, well of course they are not selling because there is no market for it. But the moment there is a market for it they split in five seconds and turn their backs to that system. Which I would say is a natural process in growing up, that disingenuousness is built in. It is so obvious and so predictable yet each generation pretends they are different. Of course every teenager that grows up thinks they are different and unique and the world doesn't know yet. As we know it’s very disappointing but (laughs). And people just grow up and make choices, they understand the world, they understand their role in the world. There are ways of going against the system and I always admire that but that's a very difficult road, so very few people choose that and they can be just as successful if they are good. So that's kind of an open source. But I don't want to diminish it because open source is an intellectual movement and it started as an intellectual idea, it’s not just a technical thing. What I see is that it gets absorbed into this whole system, that open source has a place and open source is in many ways built into the whole picture. It’s open source and it supported. Again I am totally sure that you have a ton of money to support open source so how is it different from venture capital investing in some artists? Mostly I have discussions with English artists who talk about the government saying 'they do this and they do that and erg[sic] this commercial stuff' yet if you ask them how they live and support themself it all "Arts Council”, the government that you hate is giving you a lot of money and luxuriously supports you. So that's where I think open source used to be this new phenomenon but I think it’s pretty much now built into the whole system.

MS: It’s a much harsher reality here because obviously the government support does nothing.

TB: Yeah, but if you have a good idea it’s different you will always find someone who will invest in it.

MS: Yes but are you talking about art?

TB: Well, we are talking about art specifically but you know art not the artist is open sourced. You call it art someone else calls it something else.

DS: Eyebeam has a lot of artists who work with open source, but I am getting the idea about how that is funded in a similar way to how you have just discussed, with venture capital and benefactors keeping it alive. So in a lot ways large companies sponsor open source because the browsers we use wouldn't run on a lot of the servers we use without Apache which is an open source piece of software, they appreciate it is in their better interests to support these movements as well.

MS: Yes there is a variety of motivations but the support comes in from different territories.

TB: It is more efficient for a company to support open source when they are starting research. I think it’s actually turned out pretty well. You know our crazy idea, we started this company netomat which was a very early multi media browser idea, a very ingenious idea but it wasn't really presented as a super product which would make money in five minutes. Yes in eight years people have invested ten million dollars and they still invest into it and still there is no killer application, but they still see the value and the hope. I just want to demonstrate that the whole investment thing is very rational and it pays for many different ideas. Also these people who got fabulously rich from some, you know by luck, they also support lots of crazy projects. Like ZeroOne.

MS: ZeroOne is an organisation of American curators that produce big shows and festivals out of San Jose.

TB: Deep down all these festivals are all supported by private people, organisations and cities, you know.

DS: There are often artists who will produce work and call the work and open source piece of artwork, they produce something that can be shared and they release it under various licenses such as Creative Commons licenses or the GPL license that can apply more to the actual code. Have you had any experience of work that has been licensed under those terms?

MS: I don't, it didn't come into what we show in the gallery. I think the artist that you should perhaps have some interesting insights about would be Cory, (Cory Arcangel) and that is because he hacks the non open source things and makes it available. He's one of these artists that take and give. To tell you the truth I don't know of many others within my view of the art place not the technology place that operate there.

TB: You know many of these art projects are very specific, what they create does not have a wide application.

MS: I think the specificity is not shareable.

TB: As I say more often than not these ideas are over run by others. It’s totally academic, 90 percent of public software whether it’s licensed or not because by the time it comes out something else has come out that is much better. So most software is falling aside and there is very little that carries on with some kind of volume.

DS: Yes the more artists I am talking to about this, the more successful ones seem to say “licenses? bah I just don't do it”.

TB: Exactly. It only makes money if it is applied on a massive scale and then it mostly goes on administration. So I think it is mostly moot, this issue. You know in the real world there are companies that do just that, they buy up rights in the hope that one of the thousand is actually worth something. Really it doesn't make sense. Even if you talk to a patent lawyer that's the first thing they say, just think about if it is worth it because it costs like 30 - 40 thousand dollars to get a patent and then by the time you get it it’s not worth it at all. But certainly philosophically it’s important and I think it’s important to keep up the discussion especially the part about keep extending patent shelf life, you know it’s basically driven by Mickey Mouse. In America copyright extension is driven by Mickey Mouse and I think that is very hard to learn. and I think part of it is education, people don't understand the ramifications, especially in America with the enterprise mindset to think it is always a good thing to end license a product, but it is not so beneficial.

MS: Just recently Lessig wrote an article about how incredibly against creativity the aspect of criminalisation for the new generation, how it stifles the natural impulses of making and doing things because it’s criminal. Put in those terrains of legality where a kid just doing stuff is a criminal.

TB: There is 20 or 30 years of history of major trends of appropriation, sampling all that stuff it became part of the mainstream. Yet you know the whole system is against it and tries to destroy it and unfortunately those people make a lot of money so they do everything to stifle it.

DS: The ironic thing about Disney is that the Mickey Mouse film Steamboat Willie is a rip off of Buster Keaton.

TB: It’s not about originality, it’s about having the drive, having the patent. How many times is there an obvious rip of off of somewhere, of someone who didn't patent and then someone else comes along and patents it.

MS: Well that goes a lot into commercial advertising that stuff is taken from the art world. All these patents of database films that Apple or Verizon or somebody just stole their version from Christian Marclay, the phone video. This is just one example but again there is a lot of this kind of taking because in the art world the kind of self protection paranoia does not exist. Christian in fact was advised to sue them but he didn't because he already took these pictures from all these old films. The convoluted mess of all these process and practices is too big to bother.

TB: I think it is right that you effect the next step so it is less damaging.

MS: Lessig, yes this was the most interesting conclusion.

TB: Yes, the notion that it could do that is very damaging. We used to work with a few people or associate with a few people who patented work but not really anymore.