Inscriptions of Open Source: 
between scarcity and abundance

Matthias Krauss
University of Oslo/University of Namur
2005

Word count: 13,382
Acknowledgements:

I'd like to thank my supervisors Claire Lobet and Philippe Goujon for their support and help in Belgium. A thanks to the Parallelweg girls in Maastricht for giving me a social life. And thanks to Kjersti, for her support and patience.
Synopsis:

Open Source is a phenomenon that has gained wide attention by media and academics in recent years. The motivations for contributing to open source projects has been the topic for many research projects. In the following the way different scientific fields emphasise the importance of motivations is examined. These motivations are connected to different perceptions of what role property plays in open source. The connections are examined with the help of actor-network theory and the implications of these are discussed.

Key Words: open source, inscription, actor-network theory, property, motivations.
Table of Contents

1 Introduction .......................................................................................................................... 9
2 Background and Basics of Open Source ............................................................................. 11
  2.1 A word on the use of “open source” ............................................................................... 11
  2.2 History ............................................................................................................................ 11
    2.2.1 Early years .............................................................................................................. 12
    2.2.2 Unix and BSD ......................................................................................................... 13
    2.2.3 Free Software Foundation ..................................................................................... 14
      2.2.3.1 Free Software ................................................................................................. 15
    2.2.4 Linux ..................................................................................................................... 17
  2.3 Open source Initiative .................................................................................................... 17
  2.4 Open source licenses ...................................................................................................... 18
3 Current research .................................................................................................................. 19
  3.1 Motivations for contributing ......................................................................................... 19
    3.1.1 Extrinsic motivations ............................................................................................. 20
    3.1.2 Intrinsic motivations ............................................................................................. 22
  3.2 Governance .................................................................................................................... 23
  3.3 Intellectual Property ....................................................................................................... 24
4 Theory and method .............................................................................................................. 25
  4.1 Background of actor-network theory .......................................................................... 25
  4.2 Actor-network theory .................................................................................................... 26
    4.2.1 Actors ..................................................................................................................... 26
    4.2.2 Networks ................................................................................................................ 27
    4.2.3 Stabilization in the network .................................................................................... 27
    4.2.4 Symmetry ................................................................................................................ 29
    4.2.5. Translations and inscriptions ................................................................................. 30
    4.2.6 Method - Towards Open Source ............................................................................ 31
5 Open Source – scarcity, abundance and non-scarcity .......................................................... 32
  5.1 Inscriptions of Open Source ........................................................................................ 32
    5.1.1 Who is involved? ....................................................................................................... 33
    5.1.2 What is lead where? ............................................................................................... 33
  5.2 Calculativeness – the road to decisions ....................................................................... 34
    5.2.1 The calculative actors ............................................................................................ 34
    5.2.2 Non-calculativeness ............................................................................................... 36
    5.2.3 Framing of actions .................................................................................................. 37
    5.2.4 Entanglement of things and people ....................................................................... 38
  5.3 Gifts and commodities .................................................................................................. 39
  5.4 The Open Source Marketplace – scarce goods ............................................................ 40
    5.4.1 Motivations for contributing to Open Source as a Public Good ......................... 41
  5.5 The open source gift – reciprocal exchange and abundance ........................................ 44
    5.5.1 Classical gift exchange versus digital gift exchange ............................................. 44
    5.5.2 Gifts and motivations ............................................................................................. 45
  5.6 Participatory economy .................................................................................................. 46
    5.6.1 A world of infinitely reproducible goods ............................................................... 46
    5.6.2 Participation and motivations ................................................................................ 48
6 Some actor-network perspectives ........................................................................................ 49
6.1 Economic inscriptions....................................................................................................49
  6.1.1 Public goods revisited.............................................................................................50
6.2 Gift economy inscriptions..............................................................................................51
6.3 Participatory inscription.................................................................................................51
6.4 Concluding remarks.......................................................................................................52
7 Conclusion.............................................................................................................................54
References.................................................................................................................................55
1 Introduction

“Open source” covers a lot of ground. It can be a story about property, about community structures, user-innovation, democratization of information, efficiency of open source software versus proprietary software, motivations for contributions of software code, economic rationality versus altruistic behaviour, intellectual property rights and much more. Some of these explanations are in conflict, others complement each other. What they have in common is that they (to a larger or smaller degree) attempt to explain the larger open source phenomenon from a narrowed point of view. As in the case of simple economics of open source (Lerner and Tirole 2002). Here the actors of open source projects are put in an economical framework, explaining their actions and behaviors from an economic point of view. This is not exactly wrong. One may argue for or against these arguments either inside or outside the field of economics. Sometimes both explanations from inside and outside economics have strong arguments; arguments that may be contradictory, but that make sense in their individual domains. In the case of economics the basis for interaction is exchange. An exchange based on scarcity. And scarcity can be found, or created, in open source. Other explanations seek the social glue in places like abundance, or even non-scarcity. Non-scarcity is also readily available in open source; digital information in a global network of computers may be replicated an infinite amount of time without loss of quality. Very different conclusions are drawn from these different aspects of open source.

This thesis will look at how different inscriptions are based on the role of motivations have in different views of property in open source projects. It is argued that the different inscriptions are based on various configurations of the importance of scarcity, abundance and non-scarcity,
which are all aspects of open source software and production thereof. Three different views offer three different inscriptions that leads open source in diverse directions. The thesis thus sets out to answer the question *how scientific perspectives are shaping and are shaped by open source.*

Research on open source is relatively young and the need to say something substantial about it is very much present in the current debates. Few issues are regarded as “settled” and a lot of the papers that are published on the open source phenomenon are marked by ideological and political agendas. Surely, much of the energy is also spent on “strictly scientific” matters, but still are often mingled with ideological assumptions about open source.

Actor-network theory will be used to describe the different inscriptions of open source that are taking place. Sometimes ANT is very present in the text, at other times it moves to the background follows the actors in their arguments and network building. It then again emerges as some points, attempting to make the network building more explicit, trying to show the use and positioning of actors in different translations. ANT is not a singular thing, it does not give one framework, one direction. It is about similarities as well as differences (Law 1997). In the same sense, open source is not “one thing”. Many questions form around the puzzle of “what open source is”. In the following we will examine how different points of view answer this question.
2 Background and Basics of Open Source

Here a general introduction to open source will be given, looking at its history and current developments. Open source is a young phenomenon with roots that go back to the beginning of modern computer history, but it is only in recent years that it has grabbed the attention of academics and media.

2.1 A word on the use of “open source”

Free Software was the term introduced by Richard Stallman. The establishment of the Open Source Initiative was a move away from both ambiguity and ideology, as is explained in more detail below. There are many different practices both within academic papers and elsewhere, ranging from “Open Source” to “F/LOSS” (“Free/Libre Open-Source Software”), the latter used in order create a more neutral term. However, in this thesis I will use the more popular term “open source”, without letting that be a signal of my position towards Free Software or Open Source.

2.2 History

In order to get a better understanding of the open source phenomenon, it is fruitful to take a look at its history. Open source is intimately tied to the history of modern computing, with its roots going back to the 1960s.\(^1\) Much of the history of open source has been told by central actors like Richard Stallman of the Free Software Movement and other advocates of open

\(^1\) At least that is the linear and idealistic story that is told in retrospect.
source. The stories are skewed, but they offer insight on the self-image of the hackers that have been influential on the development of the open source phenomenon. The following will be a cursory and brief excursion in the history of modern computing and open source, but will shed some vital light on the background of this rapidly spreading phenomenon. In the history that follows, the focus lies on the early development of computers in the United States and the development of software from these early days to what is known as open source software today.

2.2.1 Early years

In the beginning there was no useful distinction between hardware and software, or between users and developers (S. Weber 2004). There were just machines that could be modified through the manipulation of switches and the people who performed the modifications were developers and users of the machine at the same time. The developer’s main goal was as much the tinkering in itself as it was the creation of useful applications or a new, more effective operating system (Levy 2001). In the early days of computing, in the 1950s and 60s, computers were huge, not very powerful (by todays’ standards) and very expensive. The computer manufacturer Digital Equipment Corporation (DEC) played a major part in reducing the cost of processing power. With their PDP-series they introduced an attitude towards computers that differed substantially from the elitist thinking of IBM, the major producer of computers at the time (Levy 2001). Sharing of source code and knowledge was very common amongst researchers, as it benefited everyone and there were few economic incentives to withhold information.
2.2.2 Unix and BSD

During the summer of 1969, Ken Thompson from Bell Labs wrote the operating system Unix. In one month he spent one week each to write an operating system kernel, a shell, and editor and a compiler (S. Weber 2004). Unix has, in the last thirty years, found use on a wider set of hardware than any other operating system can claim (Raymond 2001). Ken Thompson, together with Dennis Ritche, kept improving on Unix and after a presentation at the Symposium on Operating Systems Principles at Purdue University in 1973 requests for the Unix source code came flooding in to Bell Labs. By 1976 Unix had spread around the world to Australia, England and Japan. Although Unix was developed at Bell Labs and a part of AT&T, they were prohibited to enter any markets other than those involving telecommunications. So Unix could not be sold and the source code was licensed for a few hundred dollars. Professor Bob Fabry of the University of California at Berkeley was one of many people who showed interest in Unix. With the first installation of Unix at Berkeley, a central part of the history of open source begins. In 1975, Bill Joy and Chuck Haley, two graduate students at Berkeley started working on the Pascal programming system for Unix. One year later this resulted in the first “Berkeley Software Distribution” (BSD), put together by Joy. During the following year, some thirty copies were freely distributed, and with the feedback from the user community a new version, 2BSD, was released in 1978 (McKusick 1999). The increasing interest in Unix made the AT&T management aware of its potential commercial value, but they were still bound by the decree to not commercially exploit Unix. Reluctant to give away their source code, in the end it was given only to Berkeley for “research purposes” (S. Weber 2004). At this time, alot of the computers in the ARPANET, the predecessor to the Internet, were getting outdated and there was a need for a common base
that the different software could be ported to. With BSD recently having been ported to a wideange of hardware, it was chosen as the new operating system. 4.2BSD was released in 1983
featuring full TCP/IP integration, the standard communication protocol of today’s Internet. “In
a real sense 4.2BSD lies at the foundation of the Internet as we know it today.” (S. Weber
2004:35). This year AT&T were able to create a separate division called Unix Systems
Laboratory and dramatically change the license terms of Unix. By 1988 AT&T Unix was
licensed for $100,000 (S. Weber 2004). BSD Unix was based on AT&T Unix and with the
increasing license costs from AT&T, the need for a separate and free distribution grew. By
1991 a complete rewrite from scratch had led to the first “open source” operating system;
386/BSD. 2 A community formed quickly around this release, creating NetBSD, which
focused on further portability of the system. A few months later, another group created
FreeBSD which focussed on usability for less technically advanced users on the 386-
architecture. What started out as a research project in a commercial environment and
developed through a cooperation between Berkeley University and Bell Labs, had ended up as
the stand-alone, free software operating system BSD.

2.2.3 Free Software Foundation

So while BSD was quietly releasing new, free versions of Unix, Richard Stallman was making
a choice that would give a face and voice to the principles of source code sharing, Free
Software as he would call it (Stallman 1998). He came to the MIT Artifical Intelligence Lab in
1971 and quickly became part of the hacker community there. Stallman also felt that the
sharing of source code was something natural, that there shouldn’t be any unnecessary
boundaries for making a piece of software work better or do something new. Stallman refers

2 Incidentally, this was at the same time as Linus Torvalds started his work on Linux. He has later stated that he
wouldn’t have continued work on Linux had he known about the BSD release.
to one incident that is representative for the frustration hackers felt for closed source code: a printer from the firm Xerox was constantly making trouble for the developers, but a request for the source code in order to fix the problem was turned down (Williams 2002). So even if they could have fixed the problem, they were not allowed to. At MIT they had developed a different operating system than Unix, called Incompatible Time Sharing (ITS) based on the PDP-10 computer (Levy 2001). Now, by the 1980s these machines were becoming obsolete, and since the software was written in assembler (i.e. a machine native language) for the PDP-10 meant that it would die together with the computer. The modern operating systems of the era were non-free software, meaning you had to sign a nondisclosure agreement “even to get an executable copy” (Stallman 1998:54). Richard Stallman strongly believed (and still believes) that software should be “free”, meaning that anyone should have certain freedoms when it came to modifying and distributing source code and binaries. The hacker community at MIT that Stallman had been a part of had come to an end a few years earlier, most of the hackers had been hired away into positions in commercial firms. With the increasing difficulties of sharing and getting access to source code, Stallman started the GNU (GNU’s not Unix) project in order to create a completely free operating system. At this time the BSD code was still under AT&T license and was not an alternative, so in 1985 Richard Stallman founded the Free Software Foundation in order to start the work on his GNU operating system (Stallman 1998).

2.2.3.1 Free Software

The term “Free Software” is referring to the freedom that is attached to the software and makes no implications about price, selling the software is in fact part of this freedom. There are four freedoms that make out the core of Stallman’s vision:
• Freedom to run the program, for any purpose
• Freedom to modify the program to suit your needs, which means that one has access to the source code
• Freedom to redistribute copies, gratis or for a fee
• Freedom to distribute modified versions of the program, so others can benefit from your improvements

But for Stallman it was also important that the source code remained free, to keep anyone from taking the code, modify it and then release it as proprietary software. In order to ensure that the source code remained free, Stallman invented the General Public License (GPL), inverting copyright to “copyleft”. With this, all software licensed under the GPL, including improvements of the software, is guaranteed to remain free by requiring that the source code is made available. The GPL has been described as “viral” since it has this effect on all subsequent modifications. However, one is not forced to publish any modifications of source code under GPL, but if one publishes it, it must be under the same license (Stallman 1998).

With this Stallman and the FSF were taking an aggressive stance against the proprietary software market. During the 1980s they started development of the different components that make out an operating system. The completion of the system was delayed by the fact that some of the components, like the GNU Emacs editor and GNU C compiler, became very popular and took allot of time for development and debugging (Stallman 1998). By 1990 most parts of the GNU system were ready, except for the kernel, a vital part of any operating system.
2.2.4 Linux

In 1991, Linus Torvalds released the source code for the kernel of his operating system, Linux, to an Internet newsgroup (Diamond and Torvalds 2001). The response was enormous, and by the end of the year almost 100 people had joined the newgroup, many of the making contributions to the kernel code (S. Weber 2004). By combining many of the GNU tools with the Linux kernel a complete and free operating system was created. In 1994 the first official Linux, version 1.0, was released. Today Linux is recognized as the open-source flagship, it is the most successful project the open source world has seen. The way Linux was developed is also significant to the open-source movement. Both BSD Unix and the GNU system was developed by a relatively small group of developers who were more restrictive to outside contributions than is the case with Linux (Diamond and Torvalds 2001). Linux has become the prototype of large-scale open source development, a development style that went against established rules within and outside of the hacker community. Before Linux the common consensus was that at a certain level of complexity, software had to be developed in a more centralised, a priori way. Eric S. Raymond (2001) has identified several factors that are important in this development style, such as “release early, release often” and “given enough eyeballs, are bugs are shallow” (aka. Linus’ Law). With the success of Linux, open source software was growing up and was beginning to receive attention outside of the hacker milieu.

2.3 Open source Initiative

“Open Source” was invented at a meeting in Mountain View in the offices of VA research on February 3, 1998 (Raymond 1998). Open source emerged as a need to move away from the
ambigious term Free Software, but more importantly to create an image that was more compatible with the corporate world (Raymond 1998).

Richard Stallman's zealous approach was perceived by many as anti-commercial and anti-capitalistic, something “open source” was to remedy. The open source definition was a general description of what a license had to cover in order to be entitled for the name “open source” made by the Open Source Initiative (OSI). The OSI is a non-profit organization that promotes open source and manages the open source definition by deciding what can be called “open source”, and what not. The move from free software to open source was one from ideology to technical arguments. The OSI argues for the technical superiority of open source, its rapid development process and quick diffusion. There are many licenses that fall under the open source definition, with the most popular being the GPL and BSD licenses.

2.4 Open source licenses

The GPL and BSD licenses mark two main differences between open source licenses. The BSD license does not require that modifications are licensed under the BSD. It is not viral in the way that the GPL requires any code that is under the GPL to remain under the GPL. The licenses are important to open source software and communities, while the BSD represents a more pragmatic position, the GPL shows the ideological intentions behind it. The free software foundation and Richard Stallman put the freedom to use and modify software above the arguments about technological superiority.

---

3 Always having to repeat «free as in speech, not beer» can get tiring.
3 Current research

Here there will be brief review of some of the current research on open source from different academic disciplines. The purpose then is to show the wide range of research that has been conducted, but also to provide a framework of the most pressing issues that are addressed in open source research. The reviewed research covers many aspects that fall outside the scope of this thesis, but most of them are still related to the question of property in open source. It will therefore be fruitful to go through some major areas of research and then later bring them in connection with the issues at hand in this thesis.

3.1 Motivations for contributing

“To be motivated means to be moved to do something. A person who feels no impetus or inspiration to act is thus characterized as unmotivated, whereas someone who is energized or activated toward an end is considered motivated.” (Ryan and Deci 2000).

Alot of effort has been put into explaining open source contributions within a standard economic framework (Rossi 2004). The dominating question here has been on the motivations of contributors to open source projects, and how these fit into a model of economic rationality. The puzzlement has mainly been about why people invest time and energy into something that is consequently given away for free. There have been conducted a few surveys in an attempt to classify some of the motivations underlying contributions to and participation in open source projects. Most academic work that deals with the motivation of contributors divide between intrinsic and extrinsic motivations.
3.1.1 Extrinsic motivations

Extrinsic motivations refer to an action that has a separable outcome (Ryan and Deci 2000). Of the extrinsic motivations; reputation, learning and user needs are central factors.

Reputation gain is connected to the notion of open source as a gift culture, where status is determined by what you produce and then give away, as well as an exchange culture, where scarcity forms the basis of relations. Eric S. Raymond, hacker-gone-ethnographer, has been describing the open source culture as one of gift-giving, drawing on works by Marcel Mauss (1954). The argument is that since there is no scarcity in the digital world of software the mechanisms of an exchange economy become pointless, so participation in and contributions to the community become primary measures of status (Raymond 2001). The concept of a gift culture is not generally recognized outside the field of anthropology and criticism, especially from the field of economics, has been raised against this explanation (S. Weber 2004). Weber argues that there is an abundance of computer power, disk space and Internet bandwidth, but not an abundance of people’s time and energy that is put into development, thus returning the argument to one of scarcity. Lerner and Tirole (2002) again, place the concept of reputation gains in an economic cost-benefit framework, without using the notion of a gift culture at all. Foray et al. (2004) argue that reputational capital that is gained through voluntary work can again be exploited in academic science or labor markets. In their view, reputation is one of several private benefits that can be derived from the free revelation of information and knowledge. It is, however, only one of many incentives for sharing information and is part of an overall calculation were the benefits from sharing exceed the benefits from access restriction.

In addition, Weber argues that if reputation were the primary motivation for contributions, we
would see a lot more “strategic forking” were people would start new communities with themselves as leaders (S. Weber 2004). Also, the notion of reputation as a primary motivator is generally not recognized by the developers themselves (Raymond 2001, Diamond and Torvalds 2001).

The benefits from learning are often among the top motivations listed by developers in different surveys (Ghosh et al. 2001, Hertel et al. 2003, Lakhani and Wolf 2001). Since contributions to open source projects are done voluntarily, people are likely to choose tasks that are interesting and challenging, giving them an opportunity to work on problems they otherwise may not have the chance to pursue. Learning is also one of the private benefits that makes it more attractive to contribute to a collective development rather than free-riding on other peoples’ efforts (Foray et al. 2004).

*User needs* relate to the well-known “scratching an itch” expression, suggesting that development is motivated by the lack of software that fulfills some specific purpose, be it new features for existing software, a new piece of software or a new operating system like Linux. Linus Torvalds himself said: “I just wanted something that would run on my 386” (Torvalds 1999).

User needs can also lead to what von Hippel describes as horizontal user innovation networks, which work independently from manufacturers (von Hippel 2002). Within these networks, users can develop what they want, benefit from innovations developed by others and not be restricted to marketplace choices.
3.1.2 Intrinsic motivations

Intrinsic motivations are connected “to doing something because it is inherently interesting or enjoyable” (Ryan and Deci 2000:2). The intrinsic motivations have to do with such things as the pleasure of programming, the feeling of belonging to a community and ideological convictions that source code should be open and accessible. Like learning, the joy of programming generally scores high in surveys of open source developers (Ghosh et al. 2001, Hertel et al. 2003, Lakhani and Wolf 2001).

Intrinsic motivations can be divided into two subgroups: the first is based on intrinsic motivations that have to do with the mere joy of a task, were the action is reward in itself. The second has to do with the fulfillment of certain norms in themselves (Frey 1997); the same kind of motivation one finds in environmental and other voluntary organizations. Osterloh et al. (2004) argue that intrinsic motivations lie at the heart of an open-source project, but that they need to be complemented by extrinsically motivated actors in order to ensure wider success. Kollock (1999) also focuses on the intrinsic motivations in relation to virtual communities and the role of digital goods. Richard Stallman and the Free Software Foundation (FSF) have been actively focussing on the importance of the ethical and social advantages of sharing source code, as opposed to the more technically oriented arguments of the Open Source Initiative (OSI). Allowing users to cooperate and respecting user’s freedom is of primary importance (Stallman 1998) and is seen as an important intrinsic motivation for sharing source code.
3.2 Governance

The governance and organization of open source projects is another central field of academic studies. Raymond’s article “The cathedral and the bazaar” from 1998 sets out to explain the inner workings of open source and, although his metaphors have been heavily debated, set the de facto standard for describing the open source innovation process; a great babbling bazaar of differing agendas and approaches (Raymond 2001). This image is increasingly being regarded as too simplistic and inadequate to properly describe the open source development process. But the basic characteristics of the bazaar model have been followed up and divided into four subgroups: “a) the absence of a centralized decision-making unit defining ex ante the direction of development of the software code; b) concurrent design and debugging; c) the integration of users into the production of software code; d) self-selection of programmers for the tasks that best match their abilities.” (Rossi 2004:12). How an open source project is organized depends a lot on the size of the community. Many projects have few developers and don’t require any elaborate governance mechanisms. One of the most well-known large scale open-source projects is Linux and has been characterized as one were Linus Torvalds plays the role of benevolent dictator, other identified forms of coordination are rotating dictatorships and voting committees (Raymond 2001, Ljungberg 2001). Linus is known for his pragmatic and open minded style of governance, reflecting his insight that he is, after all, dependent on his followers. The very openness of the community shifts power from the leader to the followers (S. Weber 2004). Rotating dictatorship is practiced in the Perl community, were the main responsibility for the code development shifts between different core developers. The Apache community used a voting committee system up until 1996, were decisions were made through an e-mail voting system (Fielding 1999). Thus we see that the image of the anarchic bazaar is
somewhat misplaced. Holtgrewe (2004) argues that open source projects often have weaker social ties than “classic” communities of practice. She finds that this can be of advantage as the flexibility of weaker ties allows one to enter the community, but doesn’t demand that everyone is a “full member” of it. Kollock and Smith (1996) argue that the basic difficulties of cooperation continue to exist in online communities, and that the transparency and openness of online communication creates new challenges as well.

3.3 Intellectual Property

Through different licensing schemes, open source has created a notion of property that is very different from the usual proprietary software licenses. The fundamental purpose of open source licensing is to deny any the exclusive exploitation of a given work (St. Laurent 2004). Many authors have viewed the open source phenomenon as one that suggest rethinking of intellectual property protection with regard to software (Moglen 1999, Benkler 1999, Bessen 2002, K. Weber 2004). The organization of open-source projects and motivations of developers centers around the openness that is provided, in varying degree, by the open-source licenses. Having the characteristic of a public good, it is integral to the cost-benefit calculations of actors (S. Weber 2004, Foray et al. 2004). It is also important from the ethical perspective, where the sharing of information is seen as an end in itself (Moglen 1999, Stallman 1998, Zappe 2004). This notion of property, where ownership is configured around the right to distribute, rather than exclude, fits with both extrinsic and intrinsic motivations, serving the needs of a heterogeneous group that often makes out an open-source project.
4 Theory and method

4.1 Background of actor-network theory

Actor-network theory has its roots in the sociology of scientific knowledge and radical science programmes that emerged at the end of the 1960s (Asdal et al. 2001). While radical science wanted to study the economic, political and social forces that shaped the development of science and technology, the sociology of scientific knowledge went down a different path. They introduced the principle of symmetry which stated that both “successful” and “failed” science should be analyzed and explained using the same methods. However, the symmetry principle entailed a relativism that was not acceptable for the advocates of radical science. The problem was that this relativistic perspective did not apply any distinct kind of rationality to science and did not see its knowledge production as epistemologically different from other areas in society. Within the sociology of scientific knowledge, “the strong programme” was developed at the Edinburgh university, with Barry Barnes and David Bloor as pioneers (Asdal et al. 2001). In the beginning the sociology of scientific knowledge focussed on macro perspectives of the position and role of science in society, but in the middle of the 1970s a more micro oriented approach emerged. It took a closer look at the knowledge production in the laboratory through sociological and ethnomethodological studies. Bruno Latour was one of the early proponents of this approach, one that would later lead to actor-network theory. He showed that a richer and more complete picture emerged if one took in the full range of actors that were involved in the construction of scientific knowledge. This involved following the
scientist outside of the laboratory, breaking down traditional concepts of how and where scientific research took place (Asdal et al. 2001). This kind of empirical work made out a great part of the explanatory power that came from the science studies. By conducting sociological and ethnomethodological studies on researchers, ideas about (scientific) knowledge production, space and facts were shifted or broken down. Actor-network theory grew out of these lab studies and made one of its first appearances in *The Social Construction of Technological Systems* in 1987.

### 4.2 Actor-network theory

According to actor-network theory, everything derives its meaning from the relations it holds to other actors. There is nothing essential about an actor, besides the relational setup, that defines what it is. So open source as well must be seen as a result of the relations it holds to other actors. Through inscriptions and translation, different actors try to stabilize themselves and other actors by means of network relations.

#### 4.2.1 Actors

A central question is when “actorness” is achieved? Is there a crucial point where something moves from being a dispersed network to becoming a centralized actor? A simple answer to this is whether or not something is treated as an actor, meaning that its network relations have been black boxed. A human being is normally treated as an actor as long as the body is treated as an actor. If we move closer and start to examine the bodily functions, the black box is opened and we observe the relations that constitute what is otherwise known as a single actor; new actors emerge. What used to be a singular actor has now become the network relations
between heart and blood circulation, digestive system, nervous system and a number of other actors. And the same move can be made “the other way” were single individuals become a group, community or class. But at the same time it must be possible to address a network, without compressing it to the state of an actor. Is it all in the name? Are networks just bottomless recursive relations?

4.2.2 Networks

So why actor-network theory? Or maybe we cannot ignore the hyphen? As was shown in the previous paragraph, are actors networks that have stabilized into objects or subjects in the network space. In other words; actors are defined recursively in the network space, which again only consists of relations. But the formation of relations alone do not constitute physical objects or actors, nor are they autonomous. Relations alone cannot be said to be anything on its own, it needs actors that can be related to each other.

4.2.3 Stabilization in the network

A computer is a relatively stable array of relations, but it is also bits of pieces of plastic, silicon, metal and what not. The network relations (and their stability) are linked with what John Law (2002) calls the Euclidean object. The Euclidean object, which exists in Euclidean space, does so for two reasons: first its materiality and second its network relations. Law borrows his vocabulary from topology, a branch of mathematics that explores the character of objects in a given space. The argument is that objects such as computers or coffee cups are topologically multiple, they exist in both network and Euclidean space. Most of us are familiar with the three dimensional space that we inhabit daily, but what about network space? What
are its topological attributes like? Latour (1999) argues that the World Wide Web has killed the last bit of the critical cutting edge of the notion of network. What was this notion? The World Wide Web has been around for some ten years now, what was it like to be a network before the WWW days? The question is closely related to the question of the network's topological attributes. Networks had more to do with translation and transformation, while today a network is synonymous with transportation, without transformation (Latour 1999). Now, this lack of transformation in the network space is connected to the stability of the object; a chicken is a chicken as long as its “components” are in a stable functional relationship. This opens up for the question of whether networks depend more on its material components or vice versa? The answer that John Law provides, is that they are in a reciprocal relation, more precisely: a network shape is unbroken only as long as the Euclidean shape is also unbroken (Law 2002a). The emphasis here lying on unbroken and its meaning derived from topology; network relations may be bent and skewed, but can only be stressed to a certain limit.

As John Law (2002a) argues are Latours immutable mobiles examples of actors that are immobile in network space, but mobile in Euclidean space. The immobility in network space is actually what ensures the possibility of movement in Euclidean space, as the stable network relations are essential to the stability in Euclidean space. But then again stable network relations depend upon stable Euclidian relations; the interaction between the two is reciprocal (Law 2002 a).⁴ So networks tell us about differences; differences in relations that again make out different actors. The computer is stabilized through a set of relations, but the relations are reflected through the various components of the computer. Put simply, a computer is not a

---

⁴ This is, as Law argues, only one of many possible interactions. Network relations may depend on enactments in different topological systems than the Euclidian one. There is however an increasing difficulty with the visibility of this interaction. Since there are no limits to the possible rules and spaces generated by topology. A movement of a three-dimensional object through a fourth dimension is no problem in topological systems, but is difficult to grasp from a three-dimensional perspective.
chicken because they are different in both materiality and relations thereof. But if a chicken would be rearranged to have the same relational network, would it then be a computer? An organic computer? It would certainly not be a chicken anymore, but maybe all it would be is bits and pieces of chicken. To set up a certain network of relations, one must work in Euclidean (or material) space, and in order to arrange a certain network of materiality, one must work in network space. This network consists of heterogeneous actors, or more plainly: it is a heterogeneous network. The building of this heterogeneous network in the intersection between network space and material space is what John Law calls heterogeneous engineering (Law 1992) and later heterogeneity/materiality (Law 2002b).

### 4.2.4 Symmetry

All things are actors, humans and non-humans; whatever makes itself heard, or who's voice is surpressed, is an actor. ANT takes a radical stance in this with its symmetry between what otherwise is seen as belonging to nature or society or being technology. It is were ANT distances itself from social constructivist perspectives and technological determinism by not letting either have any privileged explanatory power; neither gives a more stable ground. But not letting the “natural” or “social” be a more stable framework than the other doesn’t imply that either one is especially unstable. Depending on the situation both might have “bedrock” qualities, but that does not lead us to abandoning the principle of symmetry.

“There are no situation-independent understandings, except (maybe) the understanding that there are no situation-independent understandings” (Bowers 1992).

Part of the beauty (or horror, depending on you perspective) of actor-network theory comes from this universality. It simply states that there is nothing that has an a priori greater explanatory power than the other, except (again: maybe) the understanding that there is
nothing that has an *a priori* greater explanatory power than the other.

But not letting either nature or society be a starting ground, doesn’t mean that ANT simply tries to balance between the the two, ending up as “happy medium” (Callon and Latour 1992). What actor-network theory tries to do, is describing another activity, called network building, collective things, quasi-objects, trials of force or heterogeneous engineering (Callon and Latour 1992, Law 1992).

Actor-network theory deals with simplicity as well as with complexity, but without necessarily treating them as dichotomies. This is were some of the difficulties with ANT become visible; It accepts and denies binaries at the same time. We may not take an *a priori* position when it comes to for example nature and society, but at the moment we begin to talk about the two, use them in descriptions and arguments, they are being positioned. But the positions are vague and in transition. They are moved around as needed, mixed together. This is why scallops refuse to anchor themselves to collectors and scientists have to go into negotiations rather than developing proper techniques for cultivating scallops. So scallops (or whatever) have to be brought into the discussion, in order to see what impact they may have. This is neither natural realism (in the case of scallops) nor technological determinism (in the case of technological artifacts and other things we may label as technology). Actor-network theory introduces non-human actors, but not technological determinism. Agency is neither randomly distributed nor granted to everyone and –thing (Latour 1993).

### 4.2.5. Translations and inscriptions

The concepts of translation and inscriptions are revolving around similar issues.
Both deal with the (more or less) strategic network building that leads to a stabilisation of an actor or object. Madeleine Akrich introduces the notion of inscription as the practices and uses that get inscribed in a technical object (Akrich 1992). In the following *inscription* will be used to describe the network building efforts of different actors, but we will also talk about *translations*.

4.2.6 Method - Towards Open Source

In this thesis, several academic papers have been analyzed and put in a framework of inscriptions. The papers were divided into three main groups, representing different perspectives. All the papers make some statements about open source that has broad implications, which will be described in the following.
5 Open Source – scarcity, abundance and non-scarcity

In the following we will look at three different translations of open source that emphasize different aspects of property and motivations. The concepts of scarcity, abundance and non-scarcity are highlighted differently, as is the mix of intrinsic and extrinsic motivations.

5.1 Inscriptions of Open Source

More often that not, explanations of open source are attempt to explain some general properties of humans beings or technology, rather than looking at the specific socio-technical network. Inscriptions here are seen as a performance of the arguments that are core to the discipline or tradition from which the explanation stems. While there are many aspects of any given social and technical setup that may be seen as a part of a greater, recurrent theme, there is the danger that important specificities are lost. The question of what open source “is”, easily becomes a question of what one wishes open source to be. This is the nature of the inscriptions of a socio-technical artifact, a selective process where preferred parts are put together to constitute a new whole. On the other hand, a “total” description or explanation loses as much to complexity as its result may become a naïve, positivistic attempt at showing how it “really is”, the choice seem to be one between bad and worse. The following descriptions of three different inscriptions of open source captures the necessary shortcomings of the approaches, while simultaneously showing the relative soundness of each of them. It is through the acting out of inscriptions that technology can be stabilized and black boxed, otherwise it remains a chimera(Akrich 1992).

Here then, each of the inscriptions are attempts at leading open source down a specific stabilizing path.
5.1.1 Who is involved?

There are number of relations surrounding open source. There is substantial difficulty in both identifying and limiting this number. Who are to be counted as actors, and why? What criteria determine this selection and when does it stop? A few common propositions in the ANT literature is to include every actor that makes him/her/itself heard, just follow the actor and stop writing when one has reached the end.

The actors presented here are the ones who have made themselves heard when it comes to different views of the role of property with regards to motivations and coordination of open source developers. Through different problematizations open source is lead in very different directions. The problem of selection in actor-network theory is not easily solved, but the selection here represents a continuum of different views on human motivations, the role of property and how this adds up to a consistent explanation of the open source phenomenon.

5.1.2 What is lead where?

In the same manner, the description of “open source” itself is a potentially endless endeavour, as there are always new connections arising when the network is expanded to include more actors. Indeed the notion of open source “itself” is misleading, since the meaning and definition of open source is connected to the relationship it holds to other actors. “Open source” translated by an economist for example, will lead to a configuration around scarcity. Given that there is interaction between the different actors in the network, part of the meaning is created through inscription and translation by researchers, but properties of open source
software and community structures play a role as well. Meaning is cocreated, there is not so much a starting point as there is a relational meaning between actors in the network. Thus, open source is not configured around anything before someone makes a statement about its configuration, i.e. meaning is inherent to a human context. For example, the notion of a gift culture in open source is not “there” before someone inscribes the property through network relations, nor is it “absent” in the sense that the world doesn't disappear when you close your eyes. It is not so much a matter of social constructivism versus technological determinism, but of a circulating movement between what is seen as the technical and the social, or nature and culture.

5.2 Calculativeness – the road to decisions

In order to understand the different inscription schemes that are presented here, a theoretical framework that captures the difference between market and gift relations, between commodities and gifts, needs to be set up. The different factors that add up to the analytically distinct categories of commodities and gifts are presented here, followed by a description of how the different actors use them when inscribing open source. The last approach takes a different stand, were exchange plays no part. By viewing human productivity as an intrinsically rewarding action that needs no incentives for sharing in a world of non-scarce goods. Finally an actor-network analysis will be offered of the different inscriptions.

5.2.1 The calculative actors

Economic theory operates with a calculative actor, one that can assess costs and benefits. By analysing the notion of this calculativeness, some insight in the matter of gift giving and market exchange will be given. Michel Callon lists three conditions under which the ability to
make decisions, i.e. calculativeness, is possible in the laws of the markets (Callon 1998). An actor must be able to:

1) establish a list of the possible states of the world
2) rank these states of the world
3) identify and describe the actions which allow for the production of each of the possible states of the world.

The ability to perform the calculation is either explained as being an intrinsic property of homo sapiens or the influence of a specific culture. The problem with the first assumption is that it makes too strong a statement about the calculativeness of human beings, while the latter is too weak in explaining why actors embedded in a culture may shift from one mode of calculation to another. The calculative “equipment is neither all in the brains of human beings nor all in their socio-cultural frames or their institutions.” (Callon 1998:6). The two explanations form a dichotomy that poses two closed actors against each other. The first is the homo economicus, an autonomous actor that is calculative by nature. The second is homo sociologicus whose autonomy, although influenced and embedded in a cultural frame, also relies on the stabilized and closed actor. “The thesis of over-socialization, like that of under-socialization, rests on a common hypothesis: that of the existence of a person closed in on himself – a homo clausus” (Callon 1998:8). The problem is how market coordination is maintained when there is no predictable future. The solution to the problems that the closed actor – homo clausus – gets when trying to perform calculations under extreme uncertainty, i.e. how to make any decisions if you are unable to establish a list of the possible states of the world, lies in Granovetter’s social network analysis (Granovetter 1973). Especially the notion
of network as a *configuration of ontologies*, and not simply a link between actors with established identities, is important in this context. Actors can perform calculations under extreme uncertainty because they are not closed entities within an environment, but are stabilized and defined as specific actors through their relation with other actors in the network. Here then, actor and network are only two aspects of the same thing: actors define the network through their position in it, while actors again are defined through their position in the network. The similarities to actor-networks are apparent. We see a solution to the problem of calculation under uncertainty: instead of relying on an autonomous and closed actor to perform calculations, the actor is an aspect of an actor-network and what counts for the actor is the network of direct and indirect relations (Callon 1998).

### 5.2.2 Non-calculativeness

But if acting in network analysis is synonymous with calculating, thus rendering all action calculative, then how does one account for the absence of calculation? What about the uncalculated, disinterested actions of gift giving?\(^6\)

Callon presents two distinct explanation: the first is based on the subjective experience of wittingly avoiding to introduce any elements of calculation, while the other takes an objective stance and argues that disinterestedness is an illusion. Either actors believe themselves to be altruistic, but really inscribe their actions in a network of reciprocity that transcends her, or they rely on a more primitive type of calculation where uncertainty makes gift giving the most rational action in order to maximize expected gains. The second option cannot be chosen since “it dissolves the agency in the structures and resolves the problem before posing it”

---

\(^6\) The notion of the gift that Callon uses here refers to the modern, christian ideal of selflessness or pure altruism; a pure gift. The research that has been conducted by Mauss and other anthropologists refers to different modes of calculated gift giving and exchange.
In other words, there is no actor, in the sense that actions are chosen, but only determined outcomes from structural or primitive properties. The first option, i.e. the subjective experience of actors, is also problematic since it relies on the statements of actors about their uncalculative actions. Besides whether they are truthful or not, there always exists the possibility of alternating motives of which the actor is not aware, so it does not provide a very strong explanation. Choosing a middle way between the two does not solve the problem either, since it only ends up with stating that there are some areas where calculations are performed and some where they are not, for example in family relations and love. While this approach is preferable to totally subjective or objective views, it only works as a poor negotiation, not resolving the issue of altruism versus egoism.

### 5.2.3 Framing of actions

Pierre Bourdieu offers a solution that makes the non-calculated actions of an actor calculateable to an observer. The time it usually takes between the initial gifting and the appearance of the counter-gift makes it possible to «mask the contradiction between the intended truth of the gift as a generous, free and one-way gesture, and the truth that makes it a moment in a relationship of exchange which transcends the singular acts of exchange.» (Bourdieu cited in Callon 1998). Institutional incentives ensure that generosity is recognized and is socially viable, making both the gift giving and counter gifting attractive. In this perspective then, it is the time frame between a gift and a counter gift that determines whether an action is experienced as calculative or disinterested. The closer the counter gift is in the time frame, the more calculative the giver will feel and vice versa. This explanation escapes arguments about essentialism, there is nothing inherently selfish or altruistic that determines

---

7 These are not areas that in themselves excludes any calculative behaviour, but often are examples of non-calculativeness.
calculation, what matters is the framing of the action. So Bourdieu’s concept of framing explains the emergence of uncalculated action. This extends the network analysis, which is constrained to known relations and combinations, simply by considering whether the return gift is part of the framing or not.

### 5.2.4 Entanglement of things and people

The entanglement of objects and actors is important in order to better understand how the different inscriptions of open source that are performed relate to different concepts of software as property. While the notions of calculation and non-calculation were clarified through the concept of framing, we still have to account for how the framing is performed. In economics this has been addressed with the notion of externality, which denotes all the connections and relations that are not taken into account in a market transaction (Callon 1998). In order to make a calculation one needs to stop including relations at a point and when this point is reached the distinct actors and goods involved in the calculation have been identified. All other actors outside this calculation are gathered under the label of externalities. Externalities may in turn be identified and incorporated in the frame, but by doing this there are always new relations that emerge, making the act of framing into a potential everlasting affair. The impossibility of total framing is called overflowing by Callon, a necessary effect of framing. In a market transaction there are always overflows, such as the information one can get from reverse engineering a car or a piece of software that is bought or licensed. This leads to the subject of entanglement, or rather disentanglement. In order to turn an object into a commodity, i.e. something that can be placed in a context where it has exchange value, it must be decontextualized, dissociated and detached. The more an object is disentangled, the more we move towards a market economy (Callon 1998). If an object is (too) entangled, then
the transaction of a commodity between seller and buyer cannot be completed.

5.3 Gifts and commodities

This is the point that Gregory (1982) makes when he describes gifts as the exchange of inalienable things. The indissoluble bond of a thing with its original owner is a central feature of a gifts exchange system (Mauss 2004), while the opposite is true in a commodity exchange system. The market creates a relationship between the objects that are exchanged, while gift culture establishes a relationship between the actors (Gregory 1982). A gift carries traces of the gift giver, creating a bond between the giver and the recipient and an obligation to reciprocate the gift. In a market transaction, actors are indifferent of each other, exchanging goods that are not entangled and are not bound to each other in any way after the transaction is completed. With the concepts of calculativeness, framing and entanglement, we can decide whether an exchange involves a gift or a commodity. There is nothing inherently uncalculative about a gift exchange. In order to explain uncalculated gift giving, framing introduces a time span that explains the subjective feeling of altruistic behaviour. But the notion of a pure gift, a gift that explicitly expects nothing in return has no social explanatory power. Mary Douglas says: “A gift that does nothing to enhance solidarity is a contradiction” (Douglas 2004, p. x). So the return gift is of importance in order to create a social bond and mostly enters in the calculations. The act of gift giving also entails an entanglement of the giver and the gift. The recipient has to know and acknowledge who the gift is from in order to reciprocate. However, the role of gifts and commodities are profoundly different in a digital environment, something that will be explored in detail in the following.
5.4 The Open Source Marketplace – scarce goods

From the perspective of economists, the challenge has been to show how the open source phenomenon fits within an economical framework, more specifically to show how the sharing of source code and software are part of a cost-benefit calculation that rests on scarcity. Accordingly the scarce aspects of source code and software, i.e. the limited human resources of brain power, time and energy, have been used as major elements in the inscription of open source. For while abundance of computer power and network bandwidth are readily acknowledged as well as the non-scarceness of information, it is scarceness that makes out the essential dynamics of open source. Because without contributions there would be no software in the first place, and without the right incentives, rational actors will not contribute anything unless the cost-benefit calculation delivers positive results. There are, of course, variations in the arguments that are proposed by the different researchers, and the collective label of «economists» is a rather violent black boxing of a heterogeneous group of actors. But what is common are the assumptions of the rational, cost-benefit calculating actor and the market as a coordinating instrument between actors. Calculation and coordination again is based on the scarcity of goods that are traded, a concept that does not translate unproblematically to the open and unrestricted distribution of open source software. For the descriptions of the translations and inscriptions in the following, the «economist» actor will suffice as a spokesperson.
5.4.1 Motivations for contributing to Open Source as a Public Good

In many analyses open source is viewed as a pure public good; non-excludable since no one is prevented from consuming it and non-rival since downloading and using it does not decrease the supply (S. Weber 2004, Foray et al 2004, Lerner and Tirole 2002, Bessen 2005). Public goods are seen as «abnormal» cases were market mechanisms fail (Hahnel and Albert 1990). For economists the problem with the public good is straightforward: Contributions to a public good generally cost quite alot for the single contributors, but the overall effect is quite small. Also public goods, being non-excludable and non-rival, may be enjoyed just as much without any contributions. Since this is known to everyone that is a potential contributor and consumer of the public good, the system ought to unravel backwards with less and less contributions, leading to the collapse of the public good. So by inscribing open source as a public good, economists frame it in a well known economic problem of free riding.

Rishab Aiyer Ghosh (1998) tells the story of the tribal cooking pot that is made magically non-rival. Since the cooking pot never gets empty, one will always be compensated for whatever one puts into the pot, since the consumption of free riders does not reduce the amount of stew that is available (Ghosh 1998). By contributing to an open source project, you are in practice giving away an infinite amount of copies, each with no additional value to you, but valuable to other single individuals. The trade then, is an infinite amount of your code for one copy of someone else's code, a deal that benefits everyone. However this does not solve

8 «Consuming» is not exactly correct when one speaks of software, but is part of the standard public good argument.
9 Or, given that the good didn't exist in the first place, it will never see the light of day, since no one may have any incentives to contribute in the first place.
the problem of free riding for economists, as one still could just turn to free riding and still reap the same benefits. For Steven Weber (2004) the solution lies in the positive network externalities that open source creates as an «antirival good». Antirival goods not only may be consumed by anyone, but the value of the good increases as more and more people use the software, the same network effect achieved when more and more people own telephones for example. The line of reasoning is that software is functional, software is based on standards, the more people that use the same standard as myself, the better.

In addition, some of the free riders may provide a bug report or request a clever feature. This argument is based on the assumption that a small group of developers keep working on the project, driven by a combination of intrinsic and extrinsic motivations.

Dominique Foray et al. (2004), also inscribe open source as a public good, but go in some more detail in doing so. They argue that firstly; the general assumption that contributions have a high cost is not necessarily is true, and secondly; that rewards from learning and reputation contribute to the refraining of free riding. By setting up a «simple game of contribution to a public good», motivations, actors and source code gets further inscribed, leading to formalised propositions, lemmas and proofs(Foray et al. 2004). For example, a developer gets inscribed as \(i\) and open software is \(G (n)\). Now with a set of functions and assumptions about developer behavior and preferences, the inscription of open source is a question of following a set of logical procedures. The market orientation of open source is explained by its success on a larger market, an orientation that may change the balance of incentives for producing source code, of which learning is seen as important (Foray et al. 2004). Learning can increase technical competence, thus increasing the human capital.
However there is no room for artistic expressions since «the purpose of software is not to communicate the expression of ideas and inspiration, but to command and control a machine.» (Foray et al. 2004:10). Lerner and Tirole (2002) have identified the signalling of programming abilities as a key driving force behind contributions. Reputation gains and other extrinsic motivations are often seen as keys. Furthermore they argue that the use of licenses such as the Open Source Definition has risen, while the use of GPL has declined, something that opens up for the use of proprietary software and commercial activity (Lerner and Tirole 2002). 11

In general, the inscriptions are focussed on the expectation that there should be a separable outcome from the time and energy invested. Intrinsic rewards are a part of the equation, but are not enough to add up to the whole picture, they are seen in combination with extrinsic ones. Free riders are turned from being a problem to becoming part of the equation, but the notion of the free rider rests on the assumptions that the expectations of the actors that contribute are configured around extrinsic rewards as well. The abundance of computer power and non-scarceness of source code as information are reduced to being non-significant (computer power) or a positive network externality (informational nature of software/source code). What is significance lies within the frame of the public good contribution game. The economic perspective is characterised by the anonymity and distentanglement of market relations.

11 They describe this as a license, which it technically is not. They also state that it rid itself from the «viral» clause of GPL that demanded that all software on a compilation, e.g. a CD, had to be under the GPL as well. In fact the OSD just repeats the clause from the GPL that it applies only to derivatives, not aggregations. Also, by August 2005 sourceforge had 67,835 registered open source projects, of which 46,722 were licensed under the GPL, second is the LPGL with 7699 followed by the BSD license with 4897 [www.sourceforge.net]
5.5 The open source gift – reciprocal exchange and abundance

Raymond was one of the first to describe open source as a gift culture, but gave only a sporadic account of the phenomenon. His arguments have been refined by others later (Bitzer et al. 2004, Zeitlyn 2003, Hemetsberger 2001, Bergquist & Ljungberg 2001). He regarded gift giving as a function for building up a reputation, in a setting of abundance (Raymond 2001). He shifted the focus from scarce labour to abundant computer power, and although he did not go into any depths to explain the gift culture any further he captured some important features of the gift exchange arguments: the role of abundance for contributions to open source projects. Lévi-Strauss argues that objects are scarce according to the social value that is attributed to them, it has nothing to do with physical availability (Ekeh 1974). Accordingly, abundance is also socially determined. There are different views of what exactly an open source gift culture entails, but the common move is one away from strict cost-benefit calculations towards notions of altruism and social relations.

5.5.1 Classical gift exchange versus digital gift exchange

The study of gift cultures has mainly revolved around “primitive” societies and the exchange of material objects (Bergquist & Ljungberg 2001). The cultural and social context of gift giving in the digital environment of the Internet is different due to the nature of the gift and the relationship between giver and receiver. Firstly, there is no single object that changes hand, so the mechanism of reciprocal gift giving lose its strength. Since there is not an exchange of gifts in the regular sense that an object changes hand, but rather one gift is given
to all members of the community and anyone beyond it, more emphasis is laid on the subjective feeling of gift giving. The desire to give a gift to the community and the costs associated with it are profoundly influenced by the ease of distribution on the Internet and the fact that digital goods are not diminished in doing this (Kollock 1998). Secondly, the giver and the recipient are not so closely bound together as is usual in traditional gift cultures.

5.5.2 Gifts and motivations

Bitzer et al. (2004) put special emphasis on intrinsic motivations in open source; need for a particular software solution, fun and gift culture. They see the gift giving as an important intrinsic motivation, a contribution that is connected to reputation gains, feelings of belonging to a community and reciprocal altruism. By ascribing some of the willingness to share with others in open source projects (and beyond it) to the properties of software, i.e. abundance of computer power and bandwidth as well as the non-scarce properties of information, the gift giving arguments gain in strength. Gift giving, based more on social exchange than economical exchange, is more open to this perspective because of the symbolic value attached to source code and software. Reputation gains are perceived differently within a gift context. In the more narrow economic perspective, reputation gains are a part of the cost-benefit calculation, a tool that has value both within and beyond the community (Lerner and Tirole 2002). In the gift giving context, reputation gains play a more distinct social role, were the gift is used to establish power relations and maintaining social relations (Bergquist & Ljunghberg 2001). There is a closer connection between the giver and the gift, they are more entangled. The exchange in open source communities is of a character that creates a feeling of general indebtedness (Hemetsberger 2001). This feeling contributes to the stability of contributions in the long run.
5.6 Participatory economy

The arguments for a participatory economy, take a very different approach than the schemes of commodity or gift exchanges. Here, intrinsic motivations are seen as especially important to the writing of open source code, but the sharing is based on a moral obligation that takes non-scarcity of information and the easy distribution of this information in digital form as a starting point. The research on open source as participatory economics is inspired by the works of Michael Albert and Robin Hahnel who take a radically different approach than traditional economics. In short, their theory is based on the abandonment of private property and the organization of work around the right of self-management (Albert and Hahnel 1991). Here the concept of participatory economy is limited to the argument of software being non-scarce, an informational technology, in an environment of abundant computer power and disk space. Also included under the heading of participatory economics are the arguments of Richard Stallman and Eben Moglen, as they share the same basic assumptions about the central property features of open source software.

5.6.1 A world of infinitely reproducible goods

Uli Zappe (2004) begins his arguments about the non-scarcity of software with a thought experiment: Imagine that we lived in a world were we can reproduce any goods imaginable, with the help of advanced technology. Now consider that a group of people have this technology (including the energy to power the reproducing technology), but lack the information necessary to produce a certain good, for example an apple. Are we acting rightly if we, the possessors of this information, decline to share this information? Through lengthy
arguments he show how classic positions of utilitarianism and formal ethical theory both lead
to the same results; it would be immoral to withhold information in a world of unlimited
reproducible goods. One of the main points is that there are no strategic advantages of
withholding information; under conditions of non-scarcity there is nothing to be gained by
doing this that couldn't be obtained otherwise (Zappe 2004). In this world of infinitely
reproducible goods, there exists no market as there exists no scarcity. What remains is the
distribution of information of whatever needs to be materialised. Social relations are not build
through exchange, but by participation of the free flow of information (Zappe 2004). This
world of unlimited reproducible goods he sees realised in cyberspace; the Internet. This is a
world were we can produce a virtually unlimited amount of copies of anything that is digital
and part of this world. Following from this, it is clear that withholding information, i.e. source
code, from others is immoral and therefore should be distributed freely amongst everyone
(Zappe 2004).

But the scarce factors; time, energy and brain power, have not been forgotten, they are just
separated and contained in their own scarce domain. Zappe sees the possibility of realising a
participatory economy on the internet and views open source as a manifestation of this
economy. This position is a reversal of the economic arguments about open source; the
possibility for free and unlimited reproduction of software is not a (beneficiary) consequence
of open source, but rather a cause for it.

12 Of course, one could withhold information for the sole purpose of gaining power, a rather questionable
motivation in a non-scarce world.
5.6.2 Participation and motivations

In the participatory scheme intrinsic motivations are all that are needed in order to make this new economy work (Zappe 2003). Moglen speaks of the “emergent property of the human mind to create” and the need for open source in order to provide the proper freedom; ownership interferes with our possibility to be creative (Moglen 1999). The playful human being – Huizinga's *homo ludens* – is used, where play denotes the absence of material interest and profit gains (Huizinga 1938). But the notion of play is here closer related to the realisation of freedom. Zappe refers to Schiller's concept of play – *Spiel* – as a foundation for intrinsic motivations as a driving force in a participatory economy. Schillers *Spiel* does not emerge from external forces, but from the free will of our imagination (Schiller in Zappe 2004). But the playfulness of imagination cannot unfold under the pressure of scarcity (Zappe 2004).

Linus Torvalds, the creator of Linux, divides human motivations into three categories: survival, social life and entertainment (Torvalds 2001). Writing open source software is part of the entertainment category, something that is “intrinsically interesting and challenging” (Torvalds 2001, p.xv).¹³

The inscriptions of the participatory economy approach presented here completely sidesteps the problems that are facing market and gift economy by moving away from the foundations of any exchange, namely scarcity. Any effort that still may have to be summoned in this scheme drowns in the unlimited reproduction and distribution of the product, the costs involved are very close to zero when everyone with a computer and connection to the internet is a recipient, or rather, a participator of the product. The move away from scarcity involves a

¹³ When asked about why he made Linux available under the GPL, he said it «felt like the natural thing to do» (Williams 2002).
separation of the information domain of the Internet from the regular market sphere, in order to provide the proper environment for *homo ludens*.

Intrinsic motivations are free from the burdens of exchange and reciprocity; the stage is set for *homo ludens*. Here then, there is a strong inscription of what open source is, and where it can lead and be lead. It is seen as a tool that will have great benefits for mankind, opening up a new era. Richard Stallman, the founder of the Free Software Foundation also holds this perspective, reminding everyone of the importance of the freedom that is undermined with proprietary software (Stallman 2002). For him, there is no question whether software should be shared or not, the very nature of digital goods overrides any other calculations.

6 Some actor-network perspectives

The inscriptions above can be analysed as follows: Using John Laws concepts of how an object is being defined and stabilized in the intersection of relations and materiality, we can see the process of this being done through inscriptions and translations. The process of translation has been described by Callon, and in the following we will follow him to some extent, namely look at the way actors build networks by negotiations.

6.1 Economic inscriptions

The scarcity of brain power is transferred to the source code, which is initially seen as something that should be kept private and secret unless the sharing of it leads to greater benefits than costs. The costs are solely connected to the work that is conducted in order to write the code. That the product can be multiplied, that source code, software, in fact anything
digital can be reproduced an infinite amount of time, is of secondary importance. But it still is important, it is part of the reasons that source code is shared, because one gets additional benefits from the positive externalities that a wide distribution provides, like the convenience that comes from telephones that are able to talk with each other. So technology plays its part, albeit only as a benefit for the single rational actor, little room is left for the concern towards others. The properties of open source, its ability to act is downplayed, the relations of the network are set up in a way that silences open source as a non-scarce object. The role that is defined for it is one of a public good, it gets translated, in the sense that it is being transformed, and is enrolled as a part of economic theory. The central actor is *homo economicus*, he is the shaper of his environment, other actors such as fellow developers and the product, the source code, are reduced to parts of his calculations. He becomes the spokes person of the open source phenomena; open source is a part of traditional economics.

6.1.1 Public goods revisited

Some criticism has been raised against the public good argument, especially its assumptions about the actors involved and the things that are contributed. Does it not matter what one shares? The same benefits that one can get from sharing software are not received from sharing music, it does not help me nearly as much if everyone is listening to Britney Spears all the time, more to the contrary. What is shared is source code, it is a list of instructions that a computer performs, a set of calculations. Whether this contribution can be used by everyone depends on what it is. Whether it is a little piece of code that returns a value between one and ten, or if it is a complete set of instructions for an operating system. It is not simply a contribution *x* that is part of project *G(n)*, it is part of a heterogeneous group of small and large contributions across a diverse set of functions and needs.
The problem of free riding has not paid enough attention to the heterogeneous needs of contributors and users. Underprovisions to a public good assumes a more homogeneous group/needs...something-something.

6.2 Gift economy inscriptions

In the gift economy perspective, more emphasis was laid on the intrinsic motivations, especially those connected to the benefits from gift giving. Costs for sharing produced source code are softened by the benefits of social relations and the feeling of altruism. The role of open source is double one, it reflects the scarcity of brain power and time, and thereby giving the gift value from a material perspective, but at the same time is the non-scarcity of software and easiness of sharing an important part of the dynamics of open source, it plays are more central part than in the traditional economic perspective. The abundant digital world and non-scarcity of information become more important actors now, as they participate in the dynamics of gift giving. The notion of open source as a gift culture plays with this dichotomy of scarcity and non-scarceness. It tries to reconcile the two, by drawing an image of human actors that are both economical and social creatures and by including the software in a more central position.

6.3 Participatory inscription

In the participatory perspective, non-scarce information in an abundant digital environment plays a central role. The issue of scarce human resources is not seen as a problem, in a sense it is the externality instead. The central position of source code, with its non-scarce properties, not only requires that it is freely shared, but also enables the human actors to freely express themselves in ways that are not possible under conditions of scarcity. Open source becomes a
liberating technology, a non-human actor that transforms previous relations that have encumbered human actors. The construct of “intellectual property” is a constraint on software that otherwise would enable the free participation of humans.

6.4 Concluding remarks

All the perspectives presented here are emphasizing on the importance of motivations, albeit with very varied weight on intrinsic and extrinsic motivations. The analysis of their inscriptions, especially the way the relations between human and non-human actors are set up differently, suggests that the scientific discipline and the object of study, e.g. economy and economics, are shaping each other (Callon 1998). Open source then, can ne seen as an object that is more than one and less than many (Law 1999). It is there as one aspect, and at the same time not. It is defined, but still escapes the bonds of definition. It is not «a […] ’property destroying’ alternative to proprietary software” (Bessen 2005) while at the same time it isn't property at all (Moglen 1999). The arguments for either gift exchange or market exchange are based on the assumption that meaning is created exclusively in the social sphere. It is solely through the effort of humans that value and meaning are created. Non-human actors are silenced and pushed out in the cold space of economic externality. There are benefits derived from the fact that the software is non-rival, non-excludable, non-scarce, but it is not counted as essential. They are also based on a notion of either/or: who is creating the software, humans or machines? The answer provided is that humans create software, not machines, and it is thus to humans we must give our full attention. After all, it is people who program the machines, without them there would be nothing in the first place (not even computers). But one could easily reverse the argument and ask how humans can create software without machines. Of course computers could have looked and behaved in a number of ways different from what
wee are used to today. But the point is not that there is something essential to computers as we know them, but rather that it is futile to regard the social and the technical as opposites. And that the social has the privilege of the “in the first place”, does not entail that there is no codependence that can arise through specific practices.
7 Conclusion

In this thesis we have looked at different inscriptions of open source. We have seen how it was shaped by and was shaping different scientific disciplines, leading to a conflict of what is to be seen as the «essence» of open source. Open source is still open for interpretation and inscriptions. In a sense it will never be truly closed, as long as it forms a dichotomy with proprietary, closed software. If open source revolutionizes as some zealous proponents suggest, then one day we will only know it as «software». But it is also open in the sense that «open source» has not been black boxed in many circles. Academics and hackers alike are debating what open source «really is», which again determines where it should go. In the last years, open source has seen a tremendous increase of interest from the commercial software world, and large companies like IBM and Sun are actively supporting the development of open source software. The phenomenon open source is so large and heterogeneous that it has room for alot of translations. The research of recent years has turned open source away from being an economical puzzle to a straightforward case of cost-benefit calculations of public goods. But it also has opened up for the notion of a change away from exchange all together. Towards a more utopian vision of human collaboration. The first tries to hold open source within the well known domains of the present, while the second draws an image of a future that might already be here, represented by the first small steps made by open source. It is unsatisfactory to simply state that the truth lies somewhere in the middle. The dynamics that have arisen in open source, through the interaction between the technical and the social is open to a wide range of inscriptions.
References


Rossi, Maria A. (2004). Decoding the “Free/Open Source (F/OSS) Software Puzzle” a survey of theoretical and empirical contributions. Universita degli Studi di Siena.


