

Licensing Freedom

An Ethical Analysis of Free and Open Source Software Licenses

Saskia van de Nieuwenhof

Supervisor: Dr. Jan Vorstenbosch
Master's Thesis in Applied Ethics
Ethics Institute
Utrecht University

January 22, 2008



© 2008, Saskia van de Nieuwenhof

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 License.

You are free:

to share: to copy, distribute and transmit the work

to remix: to adapt the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Contents

Abstract	iii
Preface	iv
1 Introduction	1
2 Copyright and copyleft	5
3 Arguments against software ownership	8
4 Freedom and software licensing schemes	14
4.1 Definitions and licensing schemes	15
4.2 Freedom in software licensing	18
5 Conclusions	24
A BSD License	27
B GNU General Public License	28
Bibliography	40

Abstract

Free and open source software enables users to use, read and modify the source code of computer programs. In proprietary software, access to the source code is not given and users are generally not permitted to use, read and modify the source code. Opponents of a proprietary system state that proprietary software is morally wrong and in this thesis, the arguments they use are evaluated.

Several licensing schemes can be used as an addition to copyright law to give users the rights described above. The most important division between these different licenses can be made on the basis of the use of the copyleft principle. This principle obliges the licensee to distribute modified versions of the software under the same, or similar terms as the original license. In this thesis, this obligation is seen as a restriction on the freedom of the individual. Comparisons and possible justifications will give better insight in this supposed contradiction.

Preface

Although this is a critical analysis of free and open source software licensing schemes, the initial idea for this thesis was born out of an interest and sympathy for the free and open source software communities. Especially their values of transparency, incrementalism¹, sharing and collaboration are highly appealing to me. All of this started within the software community, but its general ideas and methodologies are now applied to other areas. This new development especially got my attention, because these communities are the perfect combination of 21th century practical idealism, new media technology and collaboration.

After I decided to write my thesis on this topic I did an internship at the Amsterdam based think tank ‘Knowledgeland’² to gain some hands-on experience. At Knowledgeland, I was involved in the Digital Pioneers project, which supports innovative Internet projects by grass roots organisations. Within these projects the use of open source software and open content licenses are being encouraged.

Besides this, I have been involved in the development of an open source youth network called YOURCE. Our aim is to develop an online platform for a European community of young people interested in youth work and volunteering, based on open source methods. I would like to thank everybody at Knowledgeland and everybody involved in YOURCE, because they inspired and motivated me to focus on the philosophical foundations of free and open source software.

During the writing of this thesis I have moved to Edinburgh, Scotland, to accompany my boyfriend Willem during his postgraduate studies. The change of landscape, language and environment have given me both the excitement and serenity which were necessary during the process of writing. I would like to thank my love for bringing me here and for his support. I would like to thank my friend Karolien for giving comments, not only on this piece, but many of them during earlier courses.

Last, but not least, I would like to thank everybody at the Ethics Institute and my supervisor Jan Vorstenbosch; the people at the Ethics Insitute for being flexible and helpful and Jan Vorstenbosch for being so kind to accept my request for supervision at a distance and the use of unusual document formats. His extensive comments were very valuable during the writing of this thesis and their positive character kept me optimistic.

¹The idea that small players can still make useful contributions. (MULGAN ET. AL. 2005)

²Known in Dutch as ‘Kennisland’. For more information see <http://www.kennisland.nl>

Chapter 1

Introduction

Much has been said and is still to be said about our network society (CASTELLS 1996) (BENKLER 1996) (HIMANEN 2001), but what seems to be one of the most interesting points is the changing, more influential role of individuals, from consumer to producer of information. Technology can lower the thresholds for interaction, engaging and participation in society.

Software was once seen as nothing more than pieces of code which we used to give instructions to our machines. Today it enables individuals to participate and interact, with the user's location, education, language, age and ethnicity no longer as factors of importance. Since software is an enabler of political and social action we can subject it to the same normative assessment as any other political or social phenomenon. (CHOPRA & DEXTER 2006)

Free and open source software have played a major role in these developments. In contradiction to proprietary software, free and open source software enables its users to read, change and redistribute the source code. Although free and open source software share the same root, they differ in their fundamental philosophical approach to software and its importance to society as a whole. (KLANG 2005)

To give users the right to read, change and redistribute the source code, new ways of licensing software had to be found. A big area of possibilities opened up and this has led to the development of many different software distribution schemes. Although they all serve

somehow the same goal as mentioned above, these particular licenses differ in the way they give rights to their users. In this thesis we will explore how the rights and freedoms of users are being protected and preserved by the different licenses. Before reaching this point, the arguments made by the proponents of free software have to be evaluated.

Source code

First of all, it is necessary to focus on the subject of the licenses more closely. Source code is text in a certain programming language for the operation and control of computers. The code is built on a highly abstracted language with a formalized syntax and English words. In combination with symbols and punctuation, structured syntactical programs are written made up of statements, loops and conditionals. For a computer to execute this program, the source code has to be translated by a compiler into executable binary code. In comparison to the source code, this binary code is nearly impossible to understand. (BERRY 2004, p. 67)

The use of standard programming languages makes it easy for developers to read and understand each others programs. In the next lines you see the source code of a computer program in the programming language Java which prints out the text Hello World! on a display.

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println>Hello World!);
    }
}
```

Free and Open Source Software

Most software being used nowadays is proprietary, which means that the software is the property of a certain company or person. If you want to use it you have to pay for it and you will get the software in its binary form which enables your computer to use it. Consequently, users cannot see and read the source code. Subject of this thesis are the ideas of two movements who have strong objections against this closed system which disables users to read and modify the source code of software.

The first is the Free Software Foundation, founded in 1983, the second one is the Open Source Initiative, founded in 1998. Let us begin with the beginnings and see how these movements originated. Their history will make clear which objections they have against proprietary software.

One can say that it all begins with Richard Stallman, the founder and front-man of the Free Software Foundation. In 1971, Richard Stallman was working as a programmer at the MIT Artificial Intelligence Lab where he was part of a software-sharing community. Back then, sharing software was not limited to this particular community according to Stallman. Later on, in the early 1980's, sharing software became less popular, because it had become nearly impossible due to the introduction of different operating systems. Modern computers had their own operating systems and these were proprietary, which source code was not available for its users. According to Stallman, software users lost the ability to change and share software and therefore he saw and still sees the proprietary software industry as an-

tisocial and wrong.

His answer to proprietary operating systems was the development of a free operating system called GNU (pronounced as 'guh-noo'). As a recursive acronym for GNU's *Not Unix* it is a name chosen following a hacker tradition. The goal of GNU was to give its users the freedom to read, copy and adapt the source code. To comply to these goals, they needed to distribute it under certain terms and conditions to prevent GNU software from being turned into proprietary software. The method they used is called *copyleft* as a play on the term *copyright*.

Because of the growth of the GNU project, the Free Software Foundation was created in 1985. The GNU project is still not finished because, amongst other reasons, another free operating system was set up by Linus Torvalds in 1991; Linux. Within a short time Torvalds was surrounded by a big community helping him with the development of this operating system. This style of development was new and not used in the GNU project. (GAY 2002, Chapter 1)

While in the late 90's the community of users and developers of GNU/Linux systems was growing rapidly, Stallman observed a shift in their motivation and ideals. Many users were mainly interested in the software, and not so much aware of the philosophy it was based on. Some people started to use the term 'open source software' rather than 'free software', to get loose of the strong principle minded vision of the Free Software Foundation. (DIBONA ET. AL. 1999, Introduction)

Open source can be seen as a development methodology, not as a social movement, according to the Free Software Foundation. In 1998, the Open Source Initiative was set up by Bruce Perens and Eric S. Raymond. In the article *The Cathedral and the Bazaar*, Raymond describes the development of Linux as a bazaar model where the development of the code is being done in public, available to everyone on the Internet. Before Torvalds started the development of Linux, most hacker

projects were set up according to the Cathedral model; centrally planned, with a small team and no early releases. (RAYMOND 2000, p. 2) The main advantage of using the Bazaar model is the famous Linus' law which says that the more widely available the source code is for public testing and development, the more easily and quickly all problems in the software will be fixed; given enough eyeballs, all bugs are shallow. (RAYMOND 2000, p. 8)

Besides operating systems such as Linux, there are many other computer programs developed and distributed as free or open source software. The term FOSS, Free and Open Source Software, covers all these computer programs. In general, the term refers to software which can be used for any purpose and which source code is available to the user. Besides accessing the source code, the user is as well allowed to adapt and redistribute the source code under certain conditions, depending on the licensing scheme. Sometimes *Libre* is added to the term FOSS, to form FLOSS. Later on, we will see that this addition to the term is needed to cope with the ambiguity of the word 'free'.

Although functioning as an umbrella term, free and open source software do certainly not refer to one and the same thing when taking into consideration the ideas and origin of the movements as we will see in Chapters 3 and 4. To avoid further misunderstanding, referring to the term FOSS will be according definition 1.0.1.

Definition 1.0.1 (FOSS) *Free and Open Source software is software which allows the user to*

*run the program for any purpose;
study the source code and adapt it;
copy and redistribute the program and the source code.*

Central question and structure

Ethics is about what makes an act right or wrong, and what one ought to do.

Ethics is also about how to come to answer these questions and how to reason systematically about these and other moral questions. (TIMMONS 2002) The more technology plays a role in our daily life, the more moral questions will be about technological issues. The most questions are raised on the use of new technologies such as genetical modification and nanotechnology. Besides this, the major role of technology in our daily life is questioned; think about the influence of the media and the protection of our personal data. Furthermore, the most fundamental global ethical problems such as war, famine and justice cannot be separated anymore from technology and innovation. (DEVISCH & VERSCHRAEGEN 2003) Commonly in applied ethics, a moral dilemma, issue or controversy is addressed. In the case of free and open source software, such dilemmas or issues can be found, but these will not be the core of this thesis. Nevertheless, some general issues within computer ethics will be addressed such as the status of digital property and the exclusive character of copyright, to answer the central question on freedom within software licenses.

In the first part of this thesis we will discuss the main arguments in favour of free and open source software. The focus will be on the views from the side of free software, because they judge the proprietary software industry to act immoral. Therefore, the first question in this thesis will be;

With which arguments do proponents of free software, and free culture in general, argue that proprietary software is wrong?

As described shortly above, copyleft is a method to protect a work from becoming proprietary. It is developed and used in software licensing to guarantee certain freedoms to its users. A copyleft software license enables users to use, copy, adapt and redistribute a work just like an open source software license does. However, the copyleft license imposes one restriction upon its users; they cannot add any restrictions of their own.

Chapter 1. Introduction

The main subject of this thesis will be this obvious contradiction between the preservation of freedom for all, while restricting the freedom of individuals. Therefore the second part of this thesis will focus on answering the following questions;

How is the freedom of users preserved within different software licensing schemes?; and

is the restriction on licensees within copyleft licenses in contradiction with the Free Software Foundation's normative ethical standpoints?

To answer these central questions, we first need to have a more thorough understanding of the issues involved and since copyleft was developed as an answer to copyright, Chapter 2 is devoted to intellectual property, the origin of copyright and the development of copyleft.

Chapter 3 is about the fascinating and never-ending campaign the free software foundation and its supporters are holding to promote free software and free culture in general. They are using a variety of arguments which are based on historic, economic or moral grounds or a mixture of these. An analysis of these arguments give us an insight in the free software movement and therefore a better understanding of their software licensing.

In Chapter 4, the main subject of freedom and software licensing will be at stake. An introduction on the negative and positive notion of freedom as described by Berlin will lead to the assessment of the definitions of both movements. Further on, we compare software licenses on the basis of freedom for the individual. In the light of two different actions the licenses are compared; the freedom to use, read and adapt the source code and the freedom to choose how to redistribute modified versions of the software. Possible justifications are discussed and compared to the normative ethical view of the party who causes the restriction.

Chapter 5 is the concluding chapter and will represent the content written in this thesis by

clear repetition and highlighting and will finally lead us to some conclusions.

Chapter 2

Copyright and copyleft

This chapter is about copyright and copyleft and their origins. Since copyleft is the issue of concern here, a thorough history of copyright and its juridical details is not in its place in this thesis. For the further investigation into the term copyleft and its relations to freedom we need to make clear where copyright stems from and what copyleft is; in which way does it differ from copyright and why was it invented in the first place.

Copyright

Most software used is non-free software, i.e. proprietary software. Before turning to the question why software should not have owners let us first focus on the concept of intellectual property in general.

A first note on property in general is that it exists in relation to people and society. If there would be no people, an object could not be someone's property. Since property only exists within the framework of law, property is a concept which stands in relation to an object, an individual and society. (RESNIK 2003, p. 320) (HETTINGER 1989, p. 31)

To protect information, ideas and artistic works, in the early 18th century the first copyright act was used. Special legal protection was needed to protect ideas, which are non-exclusive because of their intangible¹ character. Although ideas are intangible, that what

¹Whereas in law the term *intangible* would be appropriate, in philosophy, preference is given to the term *abstract* although they do not exactly have the same meaning. An intangible property is property

is protected by copyright is only the form or manner in which an idea or information is manifested. Therefore, the actual idea, concept, style or technique is insofar covered by copyright as it is published in a certain physical form. (WIKIPEDIA 2007a) This can be paraphrased to an analogy of a bottle that is protected, not the wine. (BARLOW 1994)

In the early days, copyright was a right to the author for a limited time. When this time had expired the material fell into the public domain. (LIPINSKI & BRITZ 2000, p. 58) The public domain, also referred to as *the commons* is a place from which anyone can draw, without the permission of anyone else. Although no permission is needed from someone, nothing is said about free access. Restrictions can be applied on the public domain, but these restrictions are neutral and general and therefore guarantee equal access. (LESSIG 2005, Chapter 18)

Under the Berne Convention (which has been adopted by 163 countries), established in 1886, law on copyright got a twist, with far-reaching consequences for the public domain, by automatically giving the author of a work the copyright. From then on, as soon as the work was fixed, i.e. recorded on some physical medium, the author was entitled to all copyrights in the work. (WIKIPEDIA 2007a) Current copyright law includes a section on *fair use* which allows certain unautho-

that lacks physical existence. All objects of these kind are abstract, but the difference is that not all abstract objects can be owned. (RESNIK 2003)

rized copying and usage of material to researchers, educators and libraries. Copyright law can differ from country to country, but under the Berne Convention all works except photographic and cinematographic should be copyrighted at least 50 years after the author's death. (HETTINGER 1989, p. 34) (WIKIPEDIA 2007a)

A very important feature of intellectual property is its non-exclusive character, and it is often used as an argument by free and open source software proponents. This non-exclusiveness makes it possible for objects to be in many places at the same time and used without being consumed. One can use another person's song, recipe, computer program or dance without excluding this person from using it. Some sorts of use are being excluded, but at least, this person is able to use it for personal ends. (HETTINGER 1989, p. 34)

Another important feature in the light of this thesis is the fact that copyright gives an author exclusive rights, which means that others are prohibited to use the work without given permission by the copyright holder. Therefore copyright can be seen as a negative right, because it not only gives the author the right to use his work freely, it also prohibits other people from using the work without permission. (WIKIPEDIA 2007a)

Copyleft

Copyleft is a method using copyright law to make software 'free' and requiring all modified and extended versions of the program to be free software as well. (GAY 2002, Chapter 14) This reversion of the original term had been mentioned earlier but had for the first time officially been used for the distribution of most software developed under the GNU project in 1985. Although the term seems to embody the opposite of copyright, it actually is an *extension* of copyright, not an *alternative*. (LIANG 2004, p.24)

Where copyright says 'all rights reserved', copyleft says something different. Through an



Figure 2.1: The copyleft logo

extension of the copyright, copyleft makes it possible to distribute software under copyright which enables the user to run, copy, modify and distribute the software and source code. The only thing that makes it differ from open source software is one restriction on the above; the user may not add restrictions of his or her own, in other words; the software has to stay under the same (or similar) licensing terms. In this way, the freedoms that define 'free software' which will be discussed in Chapter 4 (p. 17), are guaranteed to everyone who has a copy.

The purpose of the Free Software Foundation was to publish GNU as free software and to accomplish this there was a need for a method. One idea was to simply put the software in the public domain without copyright. This non-proprietary system would allow users, to run, copy, modify and distribute the software. However, without any restrictions or copyrights involved, it would also allow users (or *uncooperative people*, as Stallman calls them (GAY 2002, Chapter 14)) to convert the software into proprietary software. Suppose a certain uncooperative person makes some small changes and then distributes the changed software as proprietary software. As a result, people who receive the software in its changed format do not have the same freedom that the original author gave them. (GAY 2002, Chapter 14) Stallman's conclusion for distributing software in the public domain was that many users could be reached, but middlemen could strip off the freedom.

The method of copyleft is not restricted to the use on software, similar terms can be applied to other works such as printed texts, on-

line texts or images and music.

Definition 2.0.2 (Copyleft) *Copyleft is a method, initiated by software developers, to distribute a work under terms which enables others to use, copy, adapt and redistribute the work without adding restrictions of their own; the work has to stay under the same or similar distribution terms.*

The distribution terms that are used to publish most software from the GNU project are the GNU General Public License (see Appendix B). Besides this specific software license, other licenses are available, for example the GNU Free Documentation License to use for manuals, textbooks or other documents. Wikipedia is one of the most well-known Internet communities where the GNU Free Documentation License is used. They describe their use of copyleft as follows;

Wikipedia content can be copied, modified, and redistributed *so long as* the new version grants the same freedoms to others and acknowledges the authors of the Wikipedia article used. Wikipedia articles therefore will remain free forever and can be used by anybody subject to certain restrictions, most of which serve to ensure that freedom. (WIKIPEDIA 2007b)

Contributing material to Wikipedia implicates publishing it on the Internet under the GNU Free Documentation License.

Free software doesn't imply the use of a copyleft license and since there are many licenses available from different organisations and individuals, the Free Software Foundation published a list of licenses on their website sorted on the criteria 'free software' and 'compatible with GNU General Public License'. (FREE SOFTWARE FOUNDATION 2007a)

Copyleft is an idealistic principle strongly connected to the *hacker ethic*, which consists of a set of values that form the motivation to

work as a software developer in free and open source software communities. In 2001, Finnish philosopher Pekka Himanen wrote a book on hacker ethics, where he describes it as a kind of virtue ethics. (HIMANEN 2001) Although an interesting subject, the individual motivation for hackers and developers will not be the subject here.

Chapter 3

Arguments against software ownership

The most obvious point to start when writing about the argumentation for development and use of FOSS is Richard Stallman. His crusade for free software will be the starting point for this chapter, from where other opinions will be involved as well. Important to note here is that *free* in free software refers to liberty, not price. This is expressed in their slogan; ‘free as in free speech, not as in free beer’. (FREE SOFTWARE FOUNDATION 2007b) Although free software is often distributed free of cost, this is not where ‘free’ in ‘free software’ refers to. Free software can be sold for a high price, and this is actually encouraged by the Free Software Foundation. The myth of free software as being non-commercial is thereby debunked, because there is money to earn in the distribution of free software^{1,2}

Stallman speaks and writes very passionately about why proprietary software is *wrong*. To start with, have a look at a quote from Lawrence Lessig, who is inspired by Stallman’s work and writes about the broader topic of free culture.

A free culture, like a free market, is filled with property. It is filled with

¹Most research on profits within the free and open source sector is done for a broad group of ‘open source software’ and is not specified any further. In this research the revenue from standalone open source software is predicted to grow 26 percent per year to reach \$5.8 billion in 2011 (IDC 2007).

²Not a big part of the above revenues is made on the actual selling of open source software, most of it is earned by giving companies support and training. (REID 2005)

rules of property and contract that get enforced by the state. But just as a free market is perverted if its property becomes feudal, so too can a free culture be queered by extremism in the property rights that define it. (LESSIG 2004, p. xvi)

The above quote comes from the book *Free culture* by Lawrence Lessig, which states that today’s law has a negative influence on culture and creativity. In *Free culture* an interesting connection is being drawn between today’s copyright law and the decline of culture and creativity.

Stallman has written several pieces on why software should not have owners and why software should be free. It will become clear, that Stallman thinks of the first as being necessary, but not sufficient for the latter. Therefore software can not be called ‘free’, solely by not having owners. What else is necessary for software to be ‘free’ will be discussed in Chapter 4.

Before we go to some particular arguments of Stallman against proprietary software, this is the place to emphasize the strong ethical judgements proponents of free software hold against proprietary software. Stallman, as founder of the Free Software Foundation holds one of the strongest positions, since he considers the distribution of non-free software an unethical act. (MOGLEN 2000, p. 2) The legal counselor of the Free Software Foundation, Eben Moglen, also sees no justification for the ownership of digital goods in general;

If I can provide to everyone all goods of intellectual value or beauty, for the same price that I can provide the first copy of those works to anyone, why is it ever moral to exclude anyone from anything? If you could feed everyone on earth at the cost of baking one loaf and pressing a button, what would be the moral case for charging more for bread than some people could afford to pay? (MOGLEN 2003, p. 4)

Moglen points to the fact that copying software doesn't cost anything, in contradiction with the printing of a book, where production costs are involved. A computer program can be made available on the internet, which only involves some marginal costs, for example server costs.

Not affiliated with the free software movement, Hettinger holds one of the strongest positions against intellectual property from the side of philosophers. He argues that private intellectual property is restricting the free flow of ideas and that it enhances one person's freedom at the expense of everyone else's. (HETTINGER 1989, p. 35)

Insofar as copyrights, patents, and trade secrets have these negative effects, they are hard to justify. (HETTINGER 1989, p. 36)

By rejecting the following five arguments in favour of proprietary software, Stallman shows us why software should not have owners.

Name calling; software owners would use terminology which is traditionally used only for physical objects and therefore not suitable for describing software issues.

Exaggeration; software users saying that they are harmed or suffering economic losses from the methods FOSS users are exaggerating, because the person wanting a copy would not have paid money to the owner.

The law; software owners refer to the law as if today's law reflects an unquestionable view of morality.

Natural rights; authors claiming a special connection with the program they have written are actually claiming that they, as authors, are more important than anyone else and therefore have special rights. Since using or modifying the program by others doesn't benefit them more than the author, natural rights should not give the author the power to tell others that they can't do something.

Economics; the statement that having owners of software would lead to production of more software can be true, but nothing is said about the kind of software. Stallman argues that in this regard, the software being produced does not come up to the needs of society. (GAY 2002, chapter 4)

The method of argumentation Stallman uses here is quite weak, because of its rejection of the opposite, rather than convincing us of the proposed position. Nevertheless it is interesting to have a look at these arguments because they form the basis for the free software movement. Being not very consistent and clear, restructuring is being done by filtering the above arguments to general categories. Not only Stallman's arguments are used, others involved in the discussion are added to the restructured arguments.

The conceptual argument

Regarding the first one, name calling, Stallman seems to refer to the more general issue of *inappropriate fitting of copyright to digital technology*. The copyright system fits well with printed material, because it restricts mainly the mass producers of copies. With digital material, the case of copying is of a different kind and therefore, according to some, asks for a different solution. According to

Stallman it is the flexibility of digital technology that makes it a bad fit with copyright.

From various directions, others uttered the same concern; how could something designed to protect the author's right on a work expressed on canvas, film, tape or paper be applied on works made on computers.

Barlow, cofounder of the Electronic Frontier Foundation³ argued already back in 1994 that we should develop an entirely new set of methods to protect intellectual property in digital form;

Intellectual property law cannot be patched, retrofitted, or expanded to contain digitized expression any more than real estate law might be revised to cover the allocation of broadcasting spectrum. (BARLOW 1994, p. 2)

Barlow envisioned the vastly growing Internet soon to be a platform which would capture every expression of thought ever made. Meanwhile we know that the Internet expanded as Barlow predicted, but although many information already is or is being digitalized⁴, still not every 'expression of thought' is available online. According to Barlow, law on intellectual property doesn't fit with digital expressions because this law only protects the physical expressions of ideas. When ideas are being expressed digital, this expression is much more difficult to protect than earlier expressions in the form of paintings, plays, songs or designs. (BARLOW 1994, p. 2,3)

³The Electronic Frontier Foundation is an international non-profit advocacy and legal organization based in the United States with the stated purpose of being dedicated to preserve free speech rights. http://en.wikipedia.org/wiki/Electronic_Frontier_Foundation

⁴A fine example of this is the project 'Images for the future', where 137,000 hours of video, 22,510 hours of film, 123,900 hours of audio and 2,9 million photos from different institutions and archives in the Netherlands will be collected and digitalised to make it accessible for a big audience. <http://www.beeldenvoordetoekomst.nl>

With a somewhat different view on the subject, Lessig doesn't argue that copyright doesn't fit with digital technology, but he surely doesn't want to apply copyright law on material on the Internet either. Because the Internet is a space built on commons from the very first beginning, a system as the current law on intellectual property will certainly not bring us progress, but only control. In general he argues that the current execution of protection of property rights doesn't lead to the progression it was originally designed for. (LESSIG 2005, p. 357 - 359)

The discussed issue of inappropriate fitting of copyright to digital technology is one which is not an actual argument *for* free and open source software. It can only be used as argument against the use of copyright.

The argument from harm

The second argument in the list is exaggeration, which is in fact an argument on economic losses for individuals and is based on the *non-exclusiveness of software*. Stallman argues against the statement that software owners could be harmed by the use of FOSS. According to him, no person who bought and owns a copy of a certain program is harmed when another person is getting a free copy of the same program. Before evaluating this particular argument, the general issue of non-exclusiveness needs more explanation.

An important argument against proprietary software can be deducted from the non-exclusive character of intellectual property. Similar with the argument above, an argument against copyright, doesn't necessarily imply a positive argument for free and open source software.

With physical objects, value depends on space and time and therefore, on ownership. Space and time are less clear defined and sometimes even not needed when speaking about digital material. Besides that, scarcity is a term highly problematic to apply to information only available on the Internet.

(BARLOW 1994, p. 12, 13) (RESNIK 2003, p. 321) As explained in the previous chapter, non-exclusiveness (p. 6) refers to the characteristic of intellectual property that enables us to use the object in more than one place at the same time. Besides this, using it doesn't mean consuming it; by using the object you don't exclude anyone else from using it. When you ride a bicycle, nobody else is able to use that bicycle at the same time. When you eat the cake you bought that day, nobody else is able to eat that cake anymore. But when you hum your favourite song, you thereby don't exclude anybody else from humming or singing the song. This goes for all sorts of intellectual property including computer programs. (HETTINGER 1989, p. 34, 35)

Stallman's argument was that software owners were exaggerating when saying they suffer harm or economic loss when other users copy a computer program. The non-exclusiveness of intellectual property does prevent a possessor from selling the property to others of course, but we can conclude that sharing intellectual property doesn't hinder *personal use*.

The argument from special rights

The next argument, arising from the viewpoint of software developers is about their special rights; writing a program would give the developer certain rights to the resulting product which results in outweighing their desires and interests to those of others. Stallman disputes this in a rather vague way with repetitive use of lines originally belonging to other arguments. First of all, Stallman says that these special rights for programmers are derived from natural rights on material objects. With reference to the non-exclusive character of software, he argues that the use of others doesn't affect the author. As a second reason he refers to the justification of rights for authors as being based on an unquestioned tradition. He tries to reject this with the statement that copyright wasn't originally developed be-

cause of special rights of authors or rewarding them.

Stallman argues the idea that authors have special rights on their work and that their interests concerning the software outweighs those of others. Hereby, he argues the libertarian view on property which has its roots in Locke's idea of nature as common until someone takes it out of this state by his labour. (LOCKE 2003, Book II, Chapter V, §27) From his ideas on the state of nature he concludes that law and government are needed solely to protect the natural rights to life, liberty and property. Restrictions by law and government on individuals are only justified if they prevent individuals from violating other's natural rights. (RESNIK 2003, p. 322) The result of this position would be a free market where government interference would be a violation of our basic moral rights. (KYMLICKA 2002, Chapter 4) Locke thought of the world as given to mankind in common (LOCKE 2003, Book II, Chapter V, §25). With this thesis, we face a great difficulty; how could anyone have property in anything if the earth is given to mankind in common? Locke states that every man is the owner of his own body. Therefore, also the products of his work (labour) is his.

Whatsoever then he removes out of the state that nature hath provided, and left it in, he hath mixed his labour with, and joined to it something of his own, and thereby makes it his property. (LOCKE 2003, Book II, Chapter V, §27)

The argument from quality of software

The last argument in the list above is the most disputable one. As a reminder, Stallman argues that the increased production of software can be ascribed to copyright, but that this does not count as an argument in favour of proprietary software because nothing is said about the quality of the software. Although it is a strong argument, quality is something under discussion and cannot be proven as easy

as the quantity of a product. This argument from the side of opponents of free software finds its roots in a utilitarian approach. If the protection of intellectual property by means of copyright law will maximize the promoted good in society, it is regarded as the right thing to do. The encouragement of authors, entrepreneurs and inventors, through the incentives and awards intellectual property rights bring to the them, will lead to the promotion of utility. (RESNIK 2003, p. 324) (HETTINGER 1989, p. 47, 48)

Copyright was once developed to function as the motivating factor for investing time, energy and money in the development of new products and techniques. Hettinger thinks that copyright does give incentives to authors and that it therefore stimulates innovation, but he is, like Stallman, skeptical about the negative side effects of copyright. The exclusive rights that copyright provides do not increase the use and availability of new and innovative products for a big audience. (HETTINGER 1989, p. 48, 49)

Lessig, too, emphasizes this *turn* in the use of copyright. Once it was developed to stimulate progress, because property would lead to progress, and therefore more property would lead to more progress and more perfectly protected property would lead to more perfectly protected progress. According to Lessig, the power to control the use of their work shifted from individuals to corporations and therefore is now being used to control and dominate and unavoidably locks down that what copyright should protect; creativity and innovation. (LESSIG 2005, p. 349) With fellow thinkers Lessig published the *Adelphi Charter on creativity, innovation and intellectual property* in which they call upon governments and the international community to adopt a number of principles to make room for sharing, innovation and creativity. These principles are based on the following assumption;

Creativity and investment should be recognised and rewarded.

The purpose of intellectual property law (such as copyright and patents) should be, now as it was in the past, to ensure both the sharing of knowledge and the rewarding of innovation. (ROYAL SOCIETY OF ARTS 2005)

As you may have noticed, there is one argument that didn't make it to the restructured list. It is the argument in which software owners refer to the law as justification of their actions. Pointing at the law as if it is 'the moral truth' is not a standpoint which is likely to survive in any debate. Although, normally, the law is formed in democratic ways, and should therefore reflect citizens' norms and embodies ethical principles, this does not imply that the law unquestionably gives direction to what is the right thing to do in a moral sense. This argument will therefore not be listed in the arguments in favor of FOSS.

A short recapitulation brings us the following arguments against a proprietary software system by opponents of free software; a conceptual argument, an argument from harm, an argument from special rights and an argument from quality of software. These arguments form the explanation of why the free software movement argues that a proprietary system is not the right one for software and/or intellectual property rights in general. On a critical note, it is necessary to look back at the use of the proprietary system by the Free Software Foundation. Free and open source software is based on an additional license above the copyright which gives users the right to use, read and adapt the source code. In Chapter 2 we have seen that Stallman decided to work with the proprietary system himself, because of shortcomings of the only available alternative; the public domain. Using the public domain would allow users to take software out of the public domain and take ownership of it and therefore turn free software into proprietary again.

The open source movement would reject

the distribution of software in the public domain for different reasons than the one stated above. With open source licenses, as an addition to copyright law, the source code is freely available, but the author is still the owner of the work. This makes it easier to cooperate with commercial partners, who would not want to see their work end up in the public domain.

An obvious question left at the end of this chapter is why only arguments from within the free software movement have been discussed and nothing is said about the opinions from within the open source community. Of course, many pieces have been written on open source software, but actually there is no comparable substantial discussion of arguments from their side. There certainly are pleas and discussions, but most of them are mainly technical and pragmatic and focus on the efficiency of code and the motivation of the developers within these communities.⁵

⁵Interesting pieces are *The Cathedral and the Bazaar* by Raymond and *Open Sources: Voices from the Revolution* by DiBona, Ockman and Stone.

Chapter 4

Freedom and software licensing schemes

To start questioning the role of freedom in the different licensing schemes, we need to have a common understanding of the term. Because of its diverse use in so many areas there is a need to choose for one definition of freedom. The separation of the term freedom in two senses, namely *negative* and *positive* freedom, as done by Berlin, is acknowledged by many writers and is useful for a better understanding of the term.

As a political theorist and philosopher, Berlin tries to tackle the central question of obedience and coercion. The need for obedience and the limits of coercion are core notions in this regard. Within these notions freedom plays a key role and therefore Berlin wants to examine these issues thoroughly. To start with, Berlin attaches a question to the negative and positive sense of freedom. First, negative freedom is involved in the answer to the following question;

What is the area within which the subject - a person or group of persons - is or should be left to do or be what he is able to do or be, without interference by other persons? (BERLIN 2002, p. 169)

Secondly, positive freedom is involved in the answer to this question;

What, or who, is the source of control or interference that can determine someone to do, or be, this

rather than that? (BERLIN 2002, p. 169)

To put it even more shorter, negative freedom is *freedom from* and positive freedom is *freedom to*.

Negative freedom

To be free in the negative sense means that you are free to do what you want without others *wilfully* interfering. If others prevent you from doing the thing you would otherwise do we could call this *coercion*. Nevertheless, not attaining your goal doesn't necessary point to a lack of freedom, in the political meaning of the word. Suppose you want to buy a car, but you can not afford it. This could be described as a lack of economic freedom. It could only be described as a lack of political freedom when this inability to buy the car would be caused by the interference of others. Here, the first difficulty with our definition appears; if negative freedom means not being interfered with by others, how far does this interference reach? (VAN DEN BRINK 2005) (BERLIN 2002)

Suppose you want to have your say on a certain matter in public, but you are interfered by others to do this. Imagine how this could happen. Interference in the most direct way could be that your mouth would be closed by a person by putting a piece of duct tape on it. Another way could be by threatening you in

case you would speak in public. At last, others could block your personal computer from accessing certain websites on the Internet. Although in this example, all three actions are wilful interferences, one can see the wideness of the possible interferences and the vagueness of the term ‘wilful’.

Yet it remains true that the freedom of some must at times be curtailed to secure the freedom of others. Upon what principle should this be done? If freedom is a sacred, untouchable value, there can be no such principle. (BERLIN 2002, p. 173)

Berlin agrees with the classical theorists that the area in which one can act freely should be limited to a certain extent by law. He also agrees with other classical theorists when they assume that there has to be a minimum area of personal freedom.

Definition 4.0.3 (Negative freedom)

The absence of willfully interference by others which would restrain one in attaining a goal.

Positive freedom

The second sense of freedom that Berlin distinguishes is positive freedom and in contradiction to the negative notion, freedom in the positive sense is a bit more difficult to grasp. According to many, negative freedom from interference supposes the positive value of autonomy. An autonomous choice is made when it is based on a rule that one would impose upon oneself. (VAN DEN BRINK 2005) To be your own master in life and decide what to do and how to do it forms the root of freedom in its positive sense. (VAN DEN BRINK 2005) (BERLIN 2002)

The relation between negative and positive freedom is never specified by Berlin and is one under debate (VEDDER 1995, p. 33); negative freedom would be the necessary component to reach positive freedom. Not being wittingly hindered by others paves the way to make autonomous choices. Because negative freedom

paves the way, i.e. gives one certain rights, this doesn’t imply that one has a duty to use these rights. The *right* to A doesn’t imply the *duty* to A. (VAN DEN BRINK 2005) This proposition could lead to a debate about paternalism where one supposes a duty to act autonomously.

Definition 4.0.4 (Positive freedom) *The possibility to act in such a way that gives control over one’s life.*

Regarding software, and FOSS in specific, freedom mainly concerns the use of the source code and the freedom to adapt and redistribute this code. Later on, it will become clear that the existence of a license always implicates a restriction on one’s actions. Since a restriction in the form of paying, agreeing or otherwise obtaining the license can be seen as interference to attain one’s goal negative freedom is the sense of freedom which is the subject here.

4.1 Definitions and licensing schemes

It is time to explore the two movements within the world of FOSS more extensively. To examine the liberties following from the different proprietary and FOSS licensing schemes we need to know which principles these licensing terms are based on. For proprietary licenses we already have a global view on that and in this section the focus will be on the definition and ideologies from both the Free Software Foundation and the Open Source Initiative. The software licenses itself don’t tell us much about the ideological background or motives from the movements. In their definitions both movements provide normative ethical guidance to the users of FOSS.

On the basis of their definitions of free and open source software and further writings it is possible to ascribe roughly some normative ethical standpoints to the movements in general. In the next part of this chapter these

ethical standpoints will be linked to the consequences for preserving liberties for software users.

The front-men of the movements themselves state very clear what values they and their movement stand for. In the Free Software Foundation these values are ethical ones, in contrast with the practical or pragmatic approach of the Open Source Initiative. The latter thinks of proprietary software as a sub-optimal solution. For the Free Software Movement proprietary software is not just a practical issue related to market, but morally wrong. (GAY 2002, Chapter 6)

Both the Free Software Movement and the Open Source Initiative have formed some sort of guidelines to set a standard for their supporters. It was only in 1996 that the Free Software Definition was formed, a three-page document which tells us when software can be called 'free' but with emphasis on the note that the criteria stated in the document need careful thought for interpretation. The Open Source Definition is set up a bit differently, because it is a closed set of rules which must be followed when distributing software as 'open source'. This difference between the two definitions is typically for the differences between the two parties.

One of the main reasons why the Open Source Initiative exists in the first place is the ambiguity of the term 'free' in English. (WILLIAMS 2002, Chapter 11) In many languages there are two different words referring to *free* as in free to move, free to express and free to think and free as in not having to pay for something. For example in French; 'libre' and 'gratuit' or in Dutch; 'vrij' and 'gratis'. The term free software became directly associated with 'zero cost' and was therefore not very popular with companies¹. Besides this ambiguity, the Free Software Foundation's non-cooperative attitude towards companies

¹As mentioned in Chapter 3 (p. 8), 'free' in free software is not related to price, although free software is often distributed free of cost.

was not appealing to some. In 1997 the term 'open source' became used as an answer to this anti-business image of free software. (DiBONA ET. AL. 1999, Introduction)

The Open Source Definition

The distribution terms of open source software must comply with the following criteria:

1. No restrictions on the redistribution of programs.
2. The program must always be accompanied by the source code.
3. Modifications and derived works are allowed.
4. No discrimination against persons or groups.
5. No restriction on use of the program in a specific field of endeavor.
6. Rights attached to the program must apply to all to whom the program is redistributed.
7. Rights attached to the program must not depend on the program's being part of a particular software distribution.
8. No restrictions should be placed on other software that is distributed along with the licensed software.
9. License must be technology-neutral (OPEN SOURCE INITIATIVE 2006)

Within the open source movement values such as cooperation and innovation play a major role. Necessary for the creation of software is sharing the source code. Without sharing the source code a program can not be tested well. By sharing it with a large group of people errors (so called 'bugs') will be found as mentioned in the first chapter: "given enough eyes, all bugs are shallow".

This way of working was initially taken over from science, where replication is necessary to

justify a hypothesis. Sharing source code facilitates creativity by the possibilities to combine programs, be inspired by other programs and programs left behind by someone could be taken over by someone else. By making the analogy with science, the open source movement preaches for the open sharing of information to enable fast development and the creation of new software. As the editors of *Open sources: voices from the open source revolution* tell us;

Open source can develop and debug new software with the speed and creativity of science. The computer industry needs the next generation of ideas that will come from Open Source development. (DIBONA ET. AL. 1999, Introduction)

Perens wrote the Debian Social Contract which later was adapted to create the Open Source Definition. Informally, he defines the rights assured in open source software in short as follows;

The right to make copies of the program, and distribute those copies.

The right to have access to the software's source code, a necessary preliminary before you can change it.

The right to make improvements to the program. (PERENS 1999)

The differences between the rights assured by open source as described above and the Free Software Definition seem hard to find. Nevertheless the most obvious, seemingly small difference is the use of the term 'right' in the Open Source Definition and the term 'freedom' in the Free Software Definition mentioned below.

The Free Software Definition

Free software refers to four kinds of freedom for the users of the software:

Freedom 0 The freedom to run the program, for any purpose.

Freedom 1 The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.

Freedom 2 The freedom to redistribute copies so you can help your neighbour.

Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this. (FREE SOFTWARE FOUNDATION 2007b)

In addition to the similarities with the Open Source Definition, another thing is remarkable about this definition: the absence of the copyleft principle. The Free Software Definition does not include this viral² principle in the document on the four freedoms although copyleft seems to reflect the Free Software Foundation's ideology the best. (CHOPRA & DEXTER 2006, p. 4) Besides this, the foundation itself argues that copyleft is the best way to guarantee the above four freedoms to users. Both copyleft and non-copyleft licenses are accepted by both movements although they both have strong arguments to choose one over another. The Open Source Initiative argues in favour of non-copyleft licenses because this enables users to 'mix' open source software with proprietary software. Hereby the non-copyleft license doesn't exclude cooperation with proprietary software for the user. This argument is one idea of how to reach technical efficiency within the software community. (BERRY 2004)

A copyleft license such as the GNU General Public License (from now on referred to as 'GNU GPL') says that anyone who redistributes the software must pass along these

²'Viral' refers to a self-replicating technique used in marketing, to promote a certain brand or product via social networks on the Internet.

same four freedoms. This makes it impossible to incorporate software under the GNU GPL with proprietary software (and some other non-copyleft licensed software as well). Although it is absent in the Free Software Definition, the copyleft principle plays a major role within the free software movement and is the main point of disagreement between them and the people who support open source software. Together with the fact that the majority of free software is distributed under copyleft licenses³, from now on we will exclude the licenses which guarantee free software, but are not copyleft from our discussion. (GAY 2002, Chapter 14)

To compare the different kinds of software licensing schemes we need to have a representative example from each category. For free software, as discussed before we will only consider copyleft licenses and in that regard the most used and original license is the GNU GPL v. 3.0⁴ and will therefore be the representative license in this case. In the case of an open source license the choice is not as easy because none of them is particularly popular or well-known. Because the possibility of so-called *weak* forms of copyleft in open source licenses, the choice has been made for a license that is clearly not copyleft. Therefore the BSD licenses are perfect, because they are acknowledged to be open source licenses, but non-copyleft. Last, for the proprietary license a license of the latest Microsoft operating system Vista will do. The complete GNU GPL v. 3.0 and BSD license can both be found in the appendices.

³60-70% of all free and open source projects are licensed under the GNU GPL. (LAFFAN 2007)

⁴In June 2007 this new version was released, following the GNU GPL v. 2 which had been used from 1991 to 2007 and therefore is the license under which most free software has been released until now.

4.2 Freedom in software licensing

A software license is a kind of contract which a user has to agree to when installing the program on a computer. Most licenses are used to give the user permission to do a certain thing. For example a driving license permits you to drive a car. (KLANG 2005) Software licenses permit users to do different things with software, depending on the licensing scheme. FOSS licenses permit users to use, read, adapt and redistribute the software, whereas proprietary licenses only permit users to use the software. Different statements can be made about grouping these three software licensing schemes on the basis of freedom. One possibility is to put free and open source software licensing against proprietary software on the basis of providing users access to the source code; *FOSS versus proprietary software*.

Besides this ‘traditional’ grouping there is a more interesting one on the basis of freedom for the individual. One could put open source and proprietary software licenses against free software licenses on grounds of preservation of freedom for the individual against that for the community; *free software versus open source software (and proprietary software)*.

Freedom: to read, use and adapt the source code

A FOSS license, either copyleft or not, gives the licensee more freedom than a proprietary license in the sense that it enables access to the source code of the software. Without this source code, one can not get insight in the development, working and structure of a certain computer program as discussed in Chapter 1 (p. 1). A proprietary software license permits the licensee to use the program under certain conditions which are stated in the license. The excerpt from the licensing terms of Windows Vista illustrates this.

The software is licensed, not sold.
This agreement only gives you some

rights to use the software. Microsoft reserves all other rights. (...) You may not work around any technical limitations in the software; reverse engineer, decompile or disassemble the software (...); make more copies of the software than specified in this agreement (...); publish the software for others to copy; rent, lease or lend the software. (MICROSOFT)

A user who agrees to these terms is a so-called *licensee* and the one who grants the license is a *licensor*. On the one hand, a license, no matter which one, gives the licensee more freedom since he is given permission to do certain things which he wasn't allowed to do without the license. On the other hand, a license is developed in the first place to restrict people⁵. A driving license gives you the freedom to drive a car. But the fact that driving licenses exist makes you less free in your actions. If there were no such thing as a driving license I could drive a car without the license without being punished for that. To conclude; licenses in itself are legal terms to restrict an action by default. By becoming a licensee either through payment, agreeing to the terms or passing a test, you can obtain the freedom to do a certain thing. Earlier, we defined negative freedom as *the absence of willful interference by others which would restrain one in attaining a goal*. Rewording this; one is free in the negative sense to the extent that one is able to attain their goal without willfully interference by others. Stallman defines the term 'free' in free software as in *limited in its control by others* and thereby refers to Berlin's notion of negative freedom. (GAY 2002, p. 11) Leaving aside the interference by others through the existence of licenses as described above, the focus now is on the difference in licenses

⁵Note that licenses are used in specific areas only. They can be obtained in certain professions, for the operation of vehicles, as academic degrees, for state-regulated activities such as hunting and fishing and for the use of intellectual property. (WIKIPEDIA 2007c)

which let the users read and adapt the source code and which do not.

Suppose that a software user wants to learn how a certain program that he uses on his computer works. The proprietary license gives the licensee some rights to use the software. It doesn't allow the licensee to access and read the source code. We could say that these specific terms within proprietary licenses are a willful interference by the licensor which can be the cause of the licensee not attaining his goal.

Access to the source code is a necessary condition for freedom 1 of the Free Software Definition; the freedom to study how the program works, and adapt it to your needs. (FREE SOFTWARE FOUNDATION 2007b) We could combine this plea for open access to source code with the classical statement uttered by Berlin that the only true method to attain freedom is by the use of critical reason. Understanding can pave the way for freedom. The example that Berlin uses is about a schoolboy learning mathematics. Not understanding the simplest truths of mathematics makes them obstacles to the free functioning of his mind. After understanding the symbols, the functions and other basics in mathematics;

mathematical truths no longer obtrude themselves as external entities forced upon me which I must receive whether I want to or not, but as something which I now freely will in the course of the natural functioning of my own rational activity (BERLIN 2002, p. 187, 188)

Between mathematics, music (the other example Berlin uses) and software an analogy is easily made on the basis of the necessity of knowledge about respectively formulas, notes or a certain programming language. Without this knowledge the matter can never be understood, although applying, listening and using it is possible. A proprietary license doesn't give the licensee the possibility to access the

source code and this implies that the licensee is completely dependent upon the licensor. (CHEN 2006)

Freedom: to choose how to distribute modified software

Like mentioned before, software licenses cannot only be grouped by either providing access to the source code or forbidding users to read and adapt the source code. The second grouping is the subject of this paragraph; software licenses which restrict the licensee's freedom to preserve the freedom for the community as a whole versus other licenses. Proprietary licenses are not so interesting in this regard, because the source code isn't available at all to licensees.

The free software license that will function as an example here is the GNU GPL and although it is built on the concept of freedom, on one aspect non-copyleft licenses allow greater liberties than the GNU GPL; in its possibility to mix open source software with proprietary software. The following paragraph will be on this remarkable difference and its consequences for individual users.

The goal of the Free Software Foundation with the GNU GPL was to protect the four freedoms for their community, as mentioned above; the freedom to run the program for any purpose, to study how the program works and adapt it, to redistribute copies and to improve the program and release these improvements to the public. Although freedom 2 is the freedom to redistribute copies, the Free Software Foundation makes an exception here;

certain kind of rules about the manner of distributing free software are acceptable, when they don't conflict with the central freedoms. (GAY 2002, p. 44)

As an example, copyleft is introduced and the copyleft-rule would not conflict with the central freedoms, but rather protect them. Although only mentioned as an example at this

point, this is a rather controversial statement. Important to note here is that the goal of the GNU GPL is to protect these freedoms for *future* users rather than the current licensee. To do this, the GNU GPL is based on the concept of reciprocity. The freedoms that were granted to you, as user, have to be preserved and therefore you should ensure future users the same freedoms. (CHOPRA & DEXTER 2006, p. 4, 6) (KLANG 2005) In the license itself, this is formulated as follows;

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

An open source license will at first sight differ not much from a free software license; both permit the user to read and use the source code, adapt it and redistribute it. In this last action the difference between the licenses becomes apparent. Where the GNU GPL's prioritizes the community, both non-copyleft open source licenses and proprietary licenses are developed to protect the rights of individual licensees although they function in rather different ways. As discussed in Chapter 2, copyright gives the author exclusive rights; anyone who wishes to use the work has to be given permission by the copyright holder. With a non-copyleft open source license the licensee is allowed to restrict the freedom of others not by an exclusive right on the work but by the possibility to redistribute the software as proprietary software, that is, with exclusive right on the contents of the program. Hereby, future users cannot any longer read, adapt and redistribute the source code. From this, we can conclude that a proprietary license is in every case restricting others, while a non-copyleft open source license *allows* the licensee to restrict the freedom of others. The contents of a non-copyleft open source license such as the BSD license (Appendix A) is de-

scribed by Eben Moglen, the chairman of the Software Freedom Law Center, as follows;

Here is a commons. It is not defended by copyright against appropriation. Everything in the commons may be taken and put into proprietary, non-commons production as easily as it may be incorporated into commons production. We encourage people to put material into commons, and we are indifferent as to whether the appropriative use made of commons resources is proprietary, or commons-reinforcing. (MOGLEN 2003, p. 8)

The evaluation of these differences in licensing in the light of freedom stands in close relation to the moral justification of freedom. Although moral and political judgement in almost any case deals with freedom and it is often the most important justification for any policy, this does not say anything about the moral goodness of freedom as such. Besides this, freedom can be defined in several ways and therefore we could never say that freedom, no matter its definition, is morally good. (BARRY 1965, p. 136) As said before, negative freedom is the definition which we take as leading principle here and our starting point will be Lansing Pollock's rational moral theory which says that all rational persons must value their own freedom equally to the freedom of other persons. (POLLOCK 1976, p. 332 - 337) (CHOPRA & DEXTER 2006, p. 8, 9)

In the first instance, the above statement about rational agents and how they should judge about other person's freedom sounds very much like the copyleft principle as it is stated in the GNU GPL. For example, if you read and adapted the source code of certain free software and agreed to the GNU GPL, you have to use the same terms when redistributing this software. When you agree to the GNU GPL, you agree to ensure the same freedom for future users. Exactly by this agreement, licensees of the GNU GPL do not value

their own freedom equally to the freedom of others. By the terms of the GNU GPL they are restricted in their actions to give others the freedom to use, read, adapt and redistribute software. Several things can be taken into consideration, for example the fact that the restriction of one's own freedom in certain respects in order to protect one's freedom in other respects.

Before going to possible justifications for the restricting copyleft principle within the GNU GPL, it is interesting to think about the seriousness of this restriction. From the definition of negative freedom as Berlin describes it, it is not clear if everything that causes interference with a person is diminishing one's freedom in an equal grade. Many theories seem to prioritize freedom as if not even the slightest restriction could ever be justified. Certain things need to be taken into consideration and show that restrictions can be, to a higher or lower degree, important in some situations. Suppose we have a situation where a person is restricted by others in certain respects, to protect his freedom in other respects. (VEDDER 1995, p. 92, 93) An example of this is the home detention of prisoners with the use of a band around their ankle with an integrated GPS device. Their freedom is restricted by this band, but on the other hand home detention gives them the freedom to live at home and work, instead of being imprisoned.

With this in mind, we now have two starting positions on which we can build further. Pollock's freedom principle; all rational persons must value their own freedom equally to the freedom of others. But, not every restriction can be immediately considered as an equal decrease of freedom, situational aspects have to be taken into consideration.

In section 4.1 some important differences between the free software and open source movement were distinguished; the strong moral judgements by the former versus the pragmatic approach from the latter. Besides the difference in approach, their 'target

group' differs as well. Where the free software movement speaks about *all* users and refers to future users, in the Open Source Definition we only find links to programmers. Actually, the Open Source Definition concerns itself only with programmers who are keen on keeping source code freely available. (CHOPRA & DEXTER 2006, p. 17)

To come back to the GNU GPL we can conclude that this license restricts the freedom of the licensee, because the licensee cannot distribute eventual modified works under other terms than the same or similar to the ones of the GNU GPL. Is this restriction in line with the normative ethical views of the Free Software Foundation?

To answer this question, we need to have a closer look at the restriction itself. Several things are important to note here. First of all, the licensee has to distribute modified versions of the software under the same licensing terms *if* he wants to adapt and redistribute the software in the first place. The *obligation* to the distribution of modified versions only hold when the licensee adapts and wants to distribute these adaptations of the software. In all other actions, the licensee is free to do things with the software without publishing the source code to the public. Therefore, the copyleft restriction in the GNU GPL cannot be seen as paternalistic since the restriction was not embedded in the license for the good of the individual licensees, but only to protect the freedom of others. And although open source software licenses were developed to come up to the needs of software developers who were less interested in the moral side of the story, but more in the pragmatic one, the GNU GPL does not make it impossible for developers to earn money with their work. Furthermore, the copyleft restriction can hardly be seen as 'coercion', as what Berlin calls the prevention of attaining one's goals. (CHOPRA & DEXTER 2006, p. 13, 14)

So far, we have seen three points that show for what reasons the restriction on individual freedom is not so wrong as it at first seemed

to be. The aspect which does provoke most moral controversy within the GNU GPL is the fact that the copyleft principle makes it impossible to mix free software with proprietary software;

The GNU GPL does not permit incorporating your program into proprietary programs. (see Appendix B)

If a part of a program released as proprietary software is mixed with software that is licensed under the GNU GPL, the new combined program must be distributed under the terms of the GNU GPL. This viral aspect of the GNU GPL ensures the increase of free software.

From within a utilitarian framework one could find ways to justify the promotion of free software through the copyleft principle within the GNU GPL. The restriction on freedom of the licensee could be justified as it would promote freedom for the greatest number of people, if one defines freedom as the good that should be promoted. Surprisingly, the Free Software Foundation seems to use strong deontological language. (CHOPRA & DEXTER 2006, p. 22) (BERRY 2004, p. 69, 70) Take for example Stallman's explicit use of the term 'duty';

programmers have the duty to encourage others to share, redistribute, study and improve the software we write: in other words, to write *free* software. (GAY 2002, p. 121)

Stallman himself refers literally to a form of Kantian ethics, but refers to a rule which is in fact not related to Kantian ethics;

Since I do not like the consequences that result if everyone hoards information, I am required to consider it wrong for one to do so. (GAY 2002, p. 38)

The original 'golden rule' can be linked with Kantian ethics but is formed in a negative

way, in contradiction with the rule as mentioned above. As well, the fact that action is based on consequences could not be related to a Kantian ethics.

Right next to these statements, which can be related to an ethics of duty, we find expressions with a utilitarian flavour. The Free Software Foundation's main goal is to *promote* the freedoms as described in the Free Software Definition. Copyleft licenses such as the GNU GPL are a method to reach this goal by increasing the amount of free software developed and used. Whereas the two normative ethical theories of deontology and utilitarianism are normally regarded as opposite positions, Stallman has no problems with using them side by side. In the same paragraph as where he talks about the duty of programmers to develop free software he expresses his interest in a utilitarian approach through reference to, amongst others, the use of a cost-benefit analysis;

The only way to judge is to see who is helped and who is hurt by recognizing owners of software, why, and how much. In other words, we should perform a cost-benefit analysis on behalf of society as a whole taking into account individual freedom as well as production of material goods. (GAY 2002, p. 121)

In the GNU GPL direct reference to the maximization of utility is given in a section on how to apply the terms of the GNU GPL to your software;

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. (Appendix B)

Therefore, we can conclude that a deontological approach is used when it comes to the duty of programmers to promote free

software, whereas the restriction on the individual licensee by means of the copyleft terms within the GNU GPL to ensure the greatest freedom for the greatest number of people is based on utilitarian grounds. (CHOPRA & DEXTER 2006, p. 23)

A theory based on the greatest freedom for all is not characteristic for a utilitarian theory; it could be the leading principle used in libertarianism, liberalism and social contract theories. The promotion of freedom by means of an action which does not follow the ground rule of greatest freedom *is* a typical consequentialistic approach. Thereby we have come back to our question on the preservation of freedom and copyleft as a restriction on the individual licensee. Two different kinds of freedom for the individual are distinguished here; the freedom to read, use and adapt the source code of computer programs and the freedom to redistribute the software under terms of one's own choice. The latter one is related to the restriction of the individual licensee to preserve the freedom for the community as whole. The Free Software Foundation acknowledges this restriction, but sees it is a necessary method to reach their goal; the promotion of free software and thereby the maximization of freedom for the greatest number of people when it comes to software. This is in line with the normative standpoint with regard to their licensing scheme; a utilitarian one in which freedom is the good to be promoted and maximized.

Chapter 5

Conclusions

This concluding chapter will give a short overview of the issues discussed in this thesis, as well as answers to the central questions as formulated in the first chapter.

The thing that distinguishes proprietary from free and open source software is the traditional use of copyright. Copyright law is a relatively new legislation, since it was only needed after the invention of the printing press. An artistic or academic work is the expression of an idea, nevertheless copyright only protects the expression of an idea, not the idea itself. Modern copyright law is in most countries connected to the Berne Convention, which implicated the end of a rich public commons, because all works became automatically copyrighted and gave the author exclusive rights. Free and open source software licenses make use of an extension of copyright which give the user the possibility to read, use and adapt the source code. Copyleft is a special case within these extensions because it requires the modified versions of the program to be distributed under similar terms. In other words, copyleft is a viral principle used within free software licenses to ensure redistributions to remain free software.

Richard Stallman's crusade for free software is built on a strong view on proprietary software as being morally wrong. Although Stallman argues that software should be free, and should not be owned, the free software community itself uses the proprietary system. The proponents of free software bring on arguments based on different grounds. Several

people argue that intellectual property rights, as they are being used nowadays, are a threat to free culture and the free flow of ideas. Besides this, the exclusive character of copyright would be unfair, because it enhances one person's freedom at the expense of everyone else's. Copyright would make a bad fit with digital technology in general, and should therefore not be used on software. Take for example copying; copyright is the right method to protect illegal copying for printed material, but doesn't seem to work so well on digital works. This is because copyright is developed to protect the expressions of ideas, and in the case of software mainly all products are being used in digital format only. Another argument against the use of copyright on software is the non-exclusive character of intellectual property in general. A piece of software can be used in more than one place at the same time by different people. This does not cause inconvenience to the other person using it at the same time. Sharing software does not hinder personal use of it. The most obvious and disputable one is the argument that copyright would function as an incentive for software developers. From a utilitarian point of view, the protection of intellectual property is a perfectly justified method if it would maximize the promoted good in society. The ones in favour of free software question this. Increased production of software tells us nothing about the quality of it and the negative effects of copyright as an exclusive right are far-reaching.

From the arguments for free software we continued in the second part of this thesis to the assessment of freedom in software licenses. Defining freedom in a negative and a positive sense as done by Berlin gives us the necessary definitions to discuss freedom within software licensing. The definitions of the Free Software Foundation and the Open Source Initiative provide guidelines for their users by explicating their values and goals. The difference between the two movements is not so clear in first instance, since they both promote software which users can use for any purpose, read the source code and adapt it to their needs. Although they aim for the same goal, the ideas and principles on which they build are fairly different. The Open Source Initiative was born out of disapproval for the non-cooperative attitude from the Free Software Foundation towards commercial parties. They left the strong moral principles of the Free Software Foundation aside, to form a group which had seen the very efficient methods used by the free software movement; working with a large group of people on a program seem to solve bugs very quickly. The Open Source Definition is built on nine rules which say that a user has the right to read, use, modify and redistribute software. The Free Software Definition consists of four freedoms, that do not differ from the things stated in the Open Source Definition, besides a different use of language. The Free Software Foundation considers copyleft as the best method to guarantee the four freedoms as described in their definition. Copyleft licenses make it impossible to mix free software with proprietary software and therefore this is the issue which makes the difference between the two movements.

Regarding the freedom for users, two kinds of groupings are made with the three kinds of software licenses; proprietary, open source and copyleft licenses. The first grouping based on the freedom to read and use the source code and the second based on the freedom to choose how to distribute modified software. In the

case of free and open source software, users are able to read, use and adapt the source code. This gives them more freedom than users who are unable to read the source code, let alone use and adapt it. The restriction in proprietary licenses, which forbids the licensee to access the source code and to do with the software whatever he likes, can be seen as an interference with the user who is for example trying to learn how the program in question works. Companies that develop proprietary software usually acknowledge this restriction, but don't see the need for a moral justification. Their first concern is probably to produce good software and to earn money with it. In this regard, in their opinion, there is no need for a moral justification for this restriction. This is exactly the point why the second paragraph on freedom is interesting, because the one who restricts licensees in that case is the Free Software Foundation whose goal it is to promote freedom amongst its users.

The one aspect in which copyleft and open source licenses differ from each other is the degree to which they give licensees the freedom to distribute modified software in the way they want to. Although the goal of the Free Software Foundation with the GNU GPL is to preserve freedom, the freedom of licensees is certainly restricted, because in case licensees want to distribute modified versions of the software they have to do this under similar terms as the GNU GPL. For users of software with a copyleft license such as the GNU GPL there is no choice to modify this software and distribute it as proprietary software, which would be possible in non-copyleft open source licenses. The viral and effective principle of copyleft within the GNU GPL is therefore a restriction on the freedom of the individual. The normative ethical background of the Free Software Foundation is scattered and references give us reasons to think about a deontological as well as a utilitarian basis. Although a big part of their ideology is strongly deontological, a utilitarian position is clearly apparent in their practice, especially in their

licenses. On utilitarian grounds, the restriction of the licensee, caused by the copyleft principle could be justified as maximizing the greatest freedom for the greatest number of people. A theory of greatest freedom would not in any case be a typical utilitarian one, but in this case it can be characterized as such, because of the use of methods which promote and maximize the greatest freedom for the greatest number of people. The main method used is the copyleft principle within licenses such as the GNU GPL and this *is* inevitably a restriction on the licensee, but justified in accordance to the Free Software Foundation's utilitarian based normative theory.

The subject of free and open source software is quite new within the area of ethics and there are several things which could be further investigated. In this thesis, the scope didn't allow the investigation of a matter which forms a strong connection to free and open source software; the moral justification of proprietary software. Another thing which has not been discussed in this thesis is the possible view on the differences in freedom within software licenses as a division of communitarianism versus libertarianism. Since free and open source software as well as open standards are more and more introduced in governmental institutions all over the world, the interest to study this subject hopefully will become more attractive and popular. Furthermore, the general proposition of a proprietary system as being morally wrong, as uttered by Stallman could be further investigated. As a matter of fact, copyright gives the author freedom in the way that he has exclusive rights over his work. This freedom stands in contrast with the freedom of others to have access to this work. This precarious relation between property, possession and liberty is important in this light.

Appendix A

BSD License

Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B

GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

Appendix B. GNU General Public License

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- (d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works

permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- (c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- (d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- (e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or

of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

Appendix B. GNU General Public License

- (a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- (b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- (c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- (d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- (e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- (f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

Appendix B. GNU General Public License

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Appendix B. GNU General Public License

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type **show w**. This is free software, and you are welcome to redistribute it under certain conditions; type **show c** for details.

The hypothetical commands **show w** and **show c** should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

Appendix B. GNU General Public License

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Bibliography

- [BARLOW 1994] Barlow, J. P. (1994). *The economy of ideas*. Wired magazine, (2.03). <http://www.wired.com/wired/archive/2.03/economy.ideas.html>.
- [BARRY 1965] Barry, B. (1965). *Political argument*. Berkeley and Los Angeles: University of California Press.
- [BENKLER 1996] Benkler, Y. (1996). *The wealth of networks, how social production transforms markets and freedom*. New Haven and London: Yale University Press. <http://www.benkler.org>.
- [BERLIN 2002] Berlin, I. (2002). *Liberty*. Oxford: Oxford University Press. Edited by Hardy, H.
- [BERRY 2004] Berry, D. M. (2004). *The contestation of code, a preliminary investigation into the discourse of the free/libre and open source movements*. Critical Discourse Studies, 1(1), 65–89.
- [VAN DEN BRINK 2005] van den Brink, B. (2005). *Vrijheid als ideaal*, ch. 2 - vrijheid heeft geen politieke kleur, p. 38–54. Amsterdam: SUN. Edited by Snels, B.
- [CASTELLS 1996] Castells, M. (1996). *The information age: economy, society and culture. Vol. I; the rise of the network society*. Oxford: Blackwell publishers.
- [CHEN 2006] Chen, S. (2006). *Free/Open source software: licensing*. Asia-Pacific Development Information program, e-primers on free/open source software. New Delhi: Elsevier. <http://www.iosn.net/licensing/foss-licensing-primer/foss-licensing-primer-release>.
- [CHOPRA & DEXTER 2006] Chopra, S. and Dexter, S. (2006). *A comparative ethical assessment of free software licensing schemes*. http://www.sci.brooklyn.cuny.edu/~sdexter/copyleft_ethics.html.
- [DEVISCH & VERSCHRAEGEN 2003] Devisch, I. and Verschraegen, G. (eds.) (2003). *De verleiding van de ethiek; over de plaats van morele argumenten in de huidige maatschappij*. Amsterdam: Uitgeverij Boom.
- [DIBONA ET. AL. 1999] DiBona, C., Ockman, S., and Stone, M. (eds.) (1999). *Open sources: voices from the open source revolution*. Sebastopol: O'Reilly, 1st ed. <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- [FREE SOFTWARE FOUNDATION 2007a] Free Software Foundation (2007a). *Compatibility of licenses*. <http://www.gnu.org/licenses/license-list.html>, accessed at August 6, 2007.

Bibliography

- [FREE SOFTWARE FOUNDATION 2007b] Free Software Foundation (2007b). *The free software definition*. <http://www.gnu.org/philosophy/free-sw.html>, accessed at October 1, 2007.
- [GAY 2002] Gay, J. (2002). *Free Software, Free Society Selected Essays of Richard M. Stallman*. Boston, MA USA: GNU Press. <http://www.gnu.org/doc/book13.html>.
- [HETTINGER 1989] Hettinger, E. C. (1989). *Justifying intellectual property*. *Philosophy and public affairs*, 18(1), 31–52.
- [HIMANEN 2001] Himanen, P. (2001). *De hacker-ethiek en de geest van het informatietijdperk*. Amsterdam: Nieuwezijds.
- [IDC 2007] IDC (2007). *Worldwide Revenue from Standalone Open Source Software Will Grow 26% to Reach \$5.8 Billion by 2011, IDC Research Indicates*. <http://www.idc.com/getdoc.jsp?containerId=prUS20711507>, accessed at January 5, 2008.
- [KLANG 2005] Klang, M. (2005). *Free software and open source: the freedom debate and its consequences*. *First Monday*, (3). http://www.firstmonday.org/issues/issue10_3/klang/index.html, accessed at October 1, 2007.
- [KYMLICKA 2002] Kymlicka, W. (2002). *Contemporary political philosophy, an introduction*. Oxford: Oxford University Press, second ed.
- [LAFFAN 2007] Laffan, L. (2007). *GPLv2 vs GPLv3; the two seminal open source licenses, their roots, consequences and repercussions*. <http://www.visionmobile.com/whitepapers.html>, accessed at January 14, 2008.
- [LESSIG 2004] Lessig, L. (2004). *Free culture: how big media uses technology and the law to lock down culture and control creativity*. New York: The penguin press. <http://www.free-culture.cc>.
- [LESSIG 2005] Lessig, L. (2005). *Perspectives on free and open source software*, ch. 18, p. 349 – 360. Cambridge, Massachusetts: The MIT Press. Edited by Feller, J. et. al, <http://mitpress.mit.edu>.
- [LIANG 2004] Liang, L. (2004). *Guide to open content licenses*. v1.2.
- [LIPINSKI & BRITZ 2000] Lipinski, T. A. and Britz, J. J. (2000). *Rethinking the ownership of information in the 21st century: ethical implications*. *Ethics and information technology*, (2), 49–71.
- [LOCKE 2003] Locke, J. (2003). *Two Treatises of Government and A Letter Concerning Toleration*. New Haven and London: Yale University Press. Edited by Shapiro, I.
- [MICROSOFT] Microsoft. *Microsoft software license terms, windows vista home basic*. <http://www.microsoft.com/about/legal/useterms/>.
- [MOGLEN 2000] Moglen, E. (2000). *Free software matters: free software or open source?*. <http://moglen.law.columbia.edu/publications/lu-07.html>, accessed at December 6, 2007.

Bibliography

- [MOGLEN 2003] Moglen, E. (2003). *Freeing the mind: free software and the death of proprietary culture*. <http://moglen.law.columbia.edu/publications/maine-speech.html>, accessed at December 6, 2007.
- [MULGAN ET. AL. 2005] Mulgan, G., Steinberg, T., and Salem, O. (2005). *Wide open; open source methods and their future potential*. London: Demos.
- [OPEN SOURCE INITIATIVE 2006] Open Source Initiative (2006). *The Open Source Definition (Annotated)*. <http://www.opensource.org/docs/definition.php>, version 1.9, accessed at December 10, 2007.
- [PERENS 1999] Perens, B. (1999). *Open sources: voices from the open source revolution*, ch. The Open Source Definition. Sebastopol: O'Reilly, 1st ed. Edited by DiBona, C. et.al, <http://www.oreilly.com/catalog/opensources/book/perens.html>.
- [POLLOCK 1976] Pollock, L. (1976). *The freedom principle*. *Ethics*, 86(4), 332 – 342.
- [RAYMOND 2000] Raymond, E. S. (2000). *The Cathedral and the Bazaar*. <http://www.catb.org/~esr/writings/cathedral-bazaar>, version 3.0.
- [REID 2005] Reid, D. (2005). *Open source turns money-spinner*. http://news.bbc.co.uk/2/hi/programmes/click_online/4407742.stm, accessed at January 6, 2008.
- [RESNIK 2003] Resnik, D. B. (2003). *A pluralistic account of intellectual property*. *Journal of business ethics*, (46), 319–335.
- [ROYAL SOCIETY OF ARTS 2005] Royal Society of Arts (2005). *Adelphi Charter on creativity, innovation and intellectual property*. http://www.sitoc.biz/adelphicharter/adelphicharter_document.asp.htm.
- [TIMMONS 2002] Timmons, M. (2002). *Moral theory: an introduction*. Oxford: Rowman & Littlefield Publishers Inc.
- [VEDDER 1995] Vedder, A. (1995). *The values of freedom*. Utrecht: Aurelio Domus Artium.
- [WIKIPEDIA 2007a] Wikipedia (2007a). <http://en.wikipedia.org/wiki/Copyright>, accessed at September 25, 2007.
- [WIKIPEDIA 2007b] Wikipedia (2007b). <http://en.wikipedia.org/wiki/Wikipedia:Copyrights>, accessed at August 3, 2007.
- [WIKIPEDIA 2007c] Wikipedia (2007c). <http://en.wikipedia.org/wiki/License>, accessed at November 28, 2007.
- [WILLIAMS 2002] Williams, S. (2002). *Richard Stallman's crusade for free software*: O'Reilly. <http://www.oreilly.com/openbook/freedom>, accessed at October 1, 2007.