

I Walk the Open Road: Toward an Open Source Philosophy

by
Robert Cooksey

Master Thesis
The European Graduate School
Division of Media and Communications
2005
Thesis Adviser: Professor Dr. Wolfgang Schirmacher

Contents

Preface

The Trumping of Ethical Questions through Actualization and Confessions of the Influence of Others on the Techniques Herein.....3

Part One

An Overview of Open Source Software and Its Components in the Context of an Investigation of Open Source as a Concept Independent of Its Actualization in the Software Movement.....8

The Open Source Software Movement and Its Partisans.....8

Open Source as a Philosophical Topic.....9

The Definition of Open Source Software According to the Publications of the “Open Source Initiative”12

An Elaboration of Source Code as Related to Software and Machine Code, and the Dismissal of Its Description through Comparison to These Terms.....14

Part Two

A Philosophical Investigation of the Concepts of “Open” and “Source” and the Relations to One another In Various Forms of Their Expression in the Presence of Dasein.....16

From Source Code to Sourc(e/ing).....16

Source in Relation to Heidegger’s Origin.....18

Source as Expression in Quotidian Language as an Indication of “Catalysis” ..19

A Philosophical Description of Source as an Intensive Ordinate through a Return to Heidegger’s Origin and an Application of Deleuzian Ontology.....22

Source as the Space of Emergence into the Open of Dasein’s Being-Open-To Her Disinhibitor in the Absence of (Dis)Inhibition.....24

Open To, Open(ness), and the Open as Exposure and Appearance in the Presence of Dasein.....27

Recapitulation.....30

How a Source Becomes Open Through an Opening of its Opening To the Intensive Differences of an Inhabited Open.....31

Part Three

A Re-Examination of Source Code, Its Differentiation as Open Source Code, and an Analysis of The Open Source Definition in the Light of The Previous Philosophical Investigation.....34

Life Techniques that Overcome Ourselves37

Addendum.....39

Appendices

Appendix A

The Open Source Definition.....41

Introduction.....41

1. Free Redistribution.....41

2. Source Code.....41

3. Derived Works.....41

4. Integrity of The Author's Source Code.....	42
5. No Discrimination Against Persons or Groups.....	42
6. No Discrimination Against Fields of Endeavor.....	42
7. Distribution of License.....	42
8. License Must Not Be Specific to a Product.....	43
9. The License Must Not Restrict Other Software.....	43
Appendix B	
GNU GENERAL PUBLIC LICENSE.....	44
Preamble.....	44
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION.....	45
END OF TERMS AND CONDITIONS.....	49
How to Apply These Terms to Your New Programs.....	49
Bibliography.....	51

Preface

The Trumping of Ethical Questions through Actualization and Confessions of the Influence of Others on the Techniques Herein

The attempt to define the prospects of information technology in metaphysical terms of domination, history, or morality is doomed to failure from the outset... In one of his last works Sartre called for this kind of living together: "Instead of secrecy, openness should reign, and I can very well imagine the day when two persons will no longer keep any secrets from one another because they will have no secrets from anyone, because the subjective life will be a fact, as totally open as the objective life." ¹

The openness of a subjective life is an ethical choice when the facts might be masked. When all is in plain view, it is no longer an issue of what should be, but an issue of how something is, it passes from the realm of ethics into the realm of experience and information. While in our everyday lives much has seemingly remained mask-able in everyday experience (the "seeming" in this phrase being crucial), in the realm of computer technologies, there are humans who are challenging the state of software's openness as remaining in the ethical realm. I am not speaking about those who are campaigning for an ethics of openness in software and information, though they contribute to the environment in which this movement is seeking its steady state. I am speaking of the hackers who are revealing the secrets that lie behind the facade of interface—the programmers who reveal information as it is regardless of ethical questions, legal ramifications, and technical difficulties. These are not those who steal into the space of the secret and preserve the secret while augmenting it with their own. I

¹ Wolfgang Schirmacher, "Privacy as an Ethical Problem in the Computer Society," trans. Virginia Cutrufelli, <<http://www.egs.edu/faculty/schirmacher/privacy.html>>, June 2005.

am speaking of the hacker as revealer of systems, programs, and protocols as they can be opened to our media lives—definition number seven in *the Jargon File*: “One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.”²

No matter the motivation, the secrets of software are being opened by hackers regardless of the ethical/legal arguments for or against their right to do so. At the same time there is a movement of programmers creating software without secrets. This movement creates code in which the source code is open to all for view, use, modification, and redistribution. While there is much to be said regarding the ethics of open—or free—software/information, once the code is available, it is open for viewing and ethical arguments concerning whether it should be are trumped by the fact of its existence in the open.

It is the fact of this existence that I seek to examine. My investigation is not into the sociological significance, the ethical imperatives, the legal innovations of licensing, the organizational theory or the quality of the product code produced under the moniker of “Open Source.” I am interested in the phenomenon itself as a *virtuality*. This language of the virtual I take from Deleuze and DeLanda. I do not mean virtual in the sense of not-real, but virtual in the sense of not actual. The real that is divergently actualized. My techniques are an assemblage of language and approaches borrowed from Heidegger, Deleuze, and DeLanda. I pull from the poetical elements of Heidegger and his thoughts on *origin* and the *open*. I call upon the modeling of a Deleuze colored by readings of DeLanda. My thinking on my own work as the application of collections

2 Eric S. Raymond, *The Jargon File* <<http://www.catb.org/~esr/jargon/html/H/hacker.html>>, July, 2005.

of lived techniques and methods rather than as a methodology of using tools is strongly influenced by readings of and conversations and coursework with Wolfgang Schirmacher.

In order that this text might be more comprehensible to the broad audience for which it is intended, I will spend a few pages making explicit much of the language and conceptual forms that will be present in the forthcoming pages. This includes both philosophical and technical language as well the component conceptual apparatuses that will be present in my thinking during this questioning. While the language did not originate in my own work, in its grafting into the mesh of techniques that inform my interaction with the world, I have shaped it to my own creative use. Any issues one may have with the techniques should be attributed to my own work. Those from whom I borrow bear none of the responsibility for this work. They are innocents.

My composite technique in this work is derived from a marriage of the world, earth, (un)folding, disconcealing language of Heidegger's "The Origin of the Work of Art" and the Deleuzian language of virtuality, actuality, difference, and intensity. My reading of Deleuze is quite close to DeLanda's Deleuze as non-essential realist. I, however, think Deleuze in relation to Heidegger, rather than in relation to the realist/social-constructivist framework that DeLanda tends to utilize in his work on Deleuzian ontology. This thinking proceeds in this way: Heidegger thinks of the earth as the concealed ground of existence.

That into which the work sets itself back and which it causes to come forth in this setting back of itself we call the earth. Earth is that which comes forth and shelters... In setting up a world, the work sets forth the earth. This setting forth must be thought here in the strict sense of the word. The work moves

the earth into the open region of a world and keeps it there.³

When he writes of a setting forth, Heidegger is speaking of the unfolding of the earth via the mechanism of a work of art's revelation of some part of the earth, which allows it to be brought into the world of human life. At the risk of being flagrantly simplistic, the earth is always concealing and revealing itself as it unfolds. Like a plow, as the ground beneath is plowed over, other ground is plowed under. This (un)folded of the earth thus conceals as it disconceals. This language of disconcealing is crucial to Heidegger's understanding of truth as disconcealment (ancient Greek, *alétheia*). The disconcealing of the earth happens when the earth enters the open as it is set forth by the work. This disconcealed earth opens a new horizon in the expansion of the world of Dasein (the world being the interweaving web of meaning that is a practical a priori of human being).

DeLanda's reading of Deleuze brings us a radical, non-essentialist realism that encompasses the virtual as real. This thinking is based upon a very dynamic view of existence. Deleuze evades much of the thought that has plagued philosophy since Aristotle articulated his explications of the separation of matter and form (what Deleuze calls the "hylomorphic model"). Instead, Deleuze conceives that the virtual is real and is in distinction from the actual rather than the real. His extensive forms are realized via intensive differences as topological attractors for morphogenetic process. Intensive differences involve such things as temperature, pressure, anger, attraction, etc. which are not extensively divisible such as length. These intensive differences act as virtual reality that is actualized in matter. Matter is not an empty container into which the

³ Martin Heidegger, "The Origin of the Work of Art," In *Martin Heidegger: Basic Writings*, ed. David Farrell Krell (New York: HarperCollins, 1993), 171-2.

qualities of the virtual pour, but virtualities such as energy minima are latent in matter and are divergently actuated in various instances of matter, i.e., a soap bubble will form a sphere when realizing the virtuality of an energy minima while salt will form cubes.⁴ Deleuze's approach seeks to correct the over-emphasis upon the steady states of matter to which most of nineteenth and twentieth century science devoted its attention. He is interested in the liminal zones in which states bifurcate or attractors (singularities of intensive difference) alter. Deleuze also evades the language of essence and instead speaks of all qualities of matter as accidental. His differentiation is instead between the singular and the ordinary accidental. The extensive structures we know as phenomena emerge from intensive difference that are not due to transcendental qualities but emerge from difference immanent in matter (and I would include information).⁵

Gross approximations of the subtlety of each of these thinkers as these may be, the descriptions should help the uninitiated to comprehend from whence I am coming. The rest can be discerned from my setting up of relations and problems, applying each concept to the problems that I present in this work as I seek to tease out a way of understanding open source that is not definitive, but evades definition through setting up a questioning posture toward the phenomenon as it gives itself to the receptive. What is source code, and how can it be open?

4 Manuel DeLanda, "Deleuze and the Open Ended Becoming of the World," <<http://www.diss.sense.uni-konstanz.de/virtualitaet/DeLanda.html>>, April, 2005.

5 Gilles Deleuze, *Difference and Repetition*, trans. Paul Patton (New York: Columbia University, 1994), 208-12.

Part One

An Overview of Open Source Software and Its Components in the Context of an Investigation of Open Source as a Concept Independent of Its Actualization in the Software Movement

The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves.⁶

The Open Source Software Movement and Its Partisans

The Open Source movement has gathered much attention in newspapers, websites, books, and television in recent years. When I utilize the terminology of movement, it is not in the sense of a consciously organized, ideological vanguard; but a complex of observable phenomena that present some sense of shared becoming (more on this later when I address *source(ing)*). There are numerous projects within—and diverse appellations for—the family of projects that has organized about the creation of software (called alternatively “Free,” “Libre,” and “Open”). The discussions, arguments, and ideologies that underpin the various designations of the movement—often referentially coagulated into the name of FLOSS (Free-Libre-Open-Source-Software)—reveal many shared conceptualizations of the guiding processes of software production and maintenance within the overlapping communities (whose differences are not reconciled through this agreement). There are strongly voiced differences within the ideological leadership of the various manifestations of this drift, however the bulk of those members who write, contribute, use and popularize the software around which

6 The Open Source Initiative, “Site Index,” <<http://www.opensource.org/index.php>>, August 2005.

the movement is built, recognize a kinship in the various areas, even when arguing for political ideology or practical negotiation with the diverse mechanisms and organizational apparatuses of early twenty-first century human life on earth.

Open Source as a Philosophical Topic

I have chosen to focus upon the language of “Open Source” for a number of reasons—mostly accidental, but partially deliberate—not least of which is that the language has spoken to me in the most powerful sense of speaking from my first exposure to it. I have thought often of related concepts in the work of Nietzsche, James, Heidegger, Dewey, and Deleuze, and have had a very difficult time from my first exposure to the language to not think of these terms in a philosophical fashion. Indeed, I was thinking of open sources before I knowingly used my first piece of Open Source Software. While some argue that the language of Open Source evades an important ethical debate concerning terms such as “Free” and “Libre,” the language of “Open” may involve ontological and ethical issues much deeper than are immediately apparent. It is this call to something deeper than is apparent saying more than I can readily voice that draws me to and keeps me with the language of open source.

While “freedom” and debates concerning it have a long history in the philosophical tradition, the language and impressions surrounding language of “open (ness)” have less of a tradition (most of the small body of work being done since the late-twentieth century). Whereas the language of “liberty” is crucial to understanding the thinking of the Enlightenment in Europe and America (and its sometimes violently

ironic impact around the world), its language has lost much of its power. I will not argue why that has happened, though it would make an interesting study (and I would be surprised if it has not already been done), but thinking the *open* in a deeply engaged way is pressing to the coming century as thinking *liberty* was to the eighteenth century in Europe and the Americas. Its common use in phrases ranging from “an open mind” to “being open to” is matched by professional and academic language such as “open systems,” “open societies,” “open development models,” and “open access.” In universities, one hears the language of having an “open university” and the Massachusetts Institute of Technology now provides its OpenCourseWare to the world. I am even producing this thesis on a word processor that is part of a software office suite called “OpenOffice.” I seek to articulate *open* not as a genealogy of the term, nor as a linguistic study of the use of the word—though those techniques will be helpful—but as a virtuality that is manifesting itself in the world around us, as a present invisible element operating in the world.

Thinking *source* is pressing in this journey as well. Source is a concept too common to notice and too valuable to ignore. When I began to expand my reflections to include the work of others in this area, I found surprisingly little done on the idea of a source. *Origin* is addressed in a number of locations, perhaps most thoroughly in Heidegger’s *The Origin of the Work of Art*.

Origin here means that from which and by which something is what it is. What something is, as it is, we call its essence. The origin of something is the source of its essence.⁷

But even here, source is left calling for a full treatment. Many thinkers have moved

⁷ Heidegger, “The Origin of the Work of Art,” p.143.

about “beginnings” and “causes,” but source calls for its articulation as the space of the emergent in its emergence.

I am seeking to touch open sources with “a hammer as with a tuning fork.”⁸ To do so, I must *open to* the open and its source and the open of the source. Open Source Software is a transport mechanism into the processes of sources opening (or perhaps opens sourcing, as the case may be presented). I will seek to immerse myself deeply enough into the phenomena to detect the virtualities realized in OSS. To do so, I must open to the phenomena so that they might more readily open themselves in their unfolding disconcealment. The recorder is engaged. If the instrument is sensitive enough, it might hold open a window—cloudy and cracked as it may be—to the source of this opening.

I will begin this process of immersion into the emergent by investigating the most immediate phenomena that present themselves to which the language of “Open Source” is applied—Open Source Software.⁹ While there may be a wealth of insight to be gained through political, sociological, economic, organizational or any other theoretical, qualitative, or quantitative modeling methodology when applied to these phenomena; that is not the task of this investigation. I seek to think Open Source in a philosophical way, and the best way that I know to reach the philosophical is to begin anywhere and begin asking questions in earnest. If the questions generate models predicated on predictive functionality, then one has stopped at the level of theory. If one continues until the questions themselves begin to generate ways of thinking that are open to their own (de)construction, then we begin to approach the philosophical. As the

8 Friedrich Nietzsche, “Twilight of the Idols,” In *The Portable Nietzsche*, ed. and trans. Walter Kauffman (New York: Viking Penguin, 1976), p.466.

9 Hereafter, “OSS.”

reader will observe, the quiet, pressing import of thinking *open* presents itself inconspicuously in the common (as in the previous sentence).

The Definition of Open Source Software According to the Publications of the “Open Source Initiative”

What is described in the phrase “Open Source Software?” This has been articulated clearly in the “Open Source Definition” (See Appendix A).¹⁰ In short, the requirements are:

1. The software must be freely distributed and distributable without fee or duty without restriction.
2. The program must include source code and must allow the program to be distributed in source code and compiled form. If source code is not distributed with the program, it must be freely available from a well-publicized means (preferably downloading via the Internet). Deliberately obfuscated code is not allowed.
3. Licenses must allow modifications to the code and derived works must be distributed under same terms as the license of the original software.
4. A license may restrict source code from being distributed in modified form only if the license allows for modification to the code at build time via a mechanism called “patch files.” The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.
5. The license must not discriminate against any person or group of persons.
6. The license must not restrict anyone from making use of the program in a specific field of endeavor.
7. The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. The rights attached to the program must not depend on the program’s being part of a particular software distribution.

¹⁰ Open Source Initiative, “The Open Source Definition,”

<<http://web.archive.org/web/20011019055512/opensource.org/docs/definition.html>>, August, 2005, (Also included as Appendix A of this paper).

9. The license must not place restrictions on other software that is distributed along with the licensed software.¹¹

The language is quite specific while remaining as broadly applicable as possible within bounds. For my reasons, I will seek to condense these. Item one is fairly clear and is an anchor for later conditions. Conditions two through four concern the conditions of distribution and the condition of the code distributed. Together, these state that the program should be distributed and distributable in both source code and compiled forms and that any code derived from the originally licensed program should inherit the license of the original software. Deliberate obfuscation is forbidden and any restrictions must provide reasonable work-around capability. The remaining conditions all concern the forbiddance of discrimination based upon person, group, field of endeavor, or software environment as well as prohibiting the infection of other software that is not a derivative of the original through the instrument of the original.

OSS is software that is freely distribut(ed / able) in both source code and compiled form. Any software derived from software governed by an Open Source license must be governed by the same license as the original software. The distributed software should not be restricted to any person, community, software environment, etc. or in any way or be obfuscated so as to de facto restrict the readability of the source code without providing readily available alternatives that would render the software to a state as if it had none of these restrictions.

¹¹ For an alternative conception of licensing see the GNU General Public License in Appendix B. The GNU GPL is an example of a license strongly in the Free Software tradition.

An Elaboration of Source Code as Related to Software and Machine Code, and the Dismissal of Its Description through Comparison to These Terms

Understanding why source code is so important to OSS will help to illuminate its significance and the debates between the various communities involved in FLOSS. This requires a bit of exploration of the terminology used by software developers to describe the various forms that software might take. The first sentence of the previous paragraph opens three concepts that are crucial to understanding OSS: *software*, *source code*, and *compiled programs*.

Software is a vague term that applies to the programs and information that allow a computer to perform tasks that are not performed strictly by the hardware of the system. There is some debate whether information in the form of data should be included when using the descriptor “software.” I will take the most inclusive path in regards to this as it is becoming increasingly difficult to discern whether data is at work or merely being processed in the practice of contemporary computing. Software is often divided into system and application software, but that distinction is not pertinent here.¹²

Software is written in a computer language that is most often compiled, interpreted, or assembled into a form that is executable (or *run*) on a computer. The general description of the process usually runs something like this: *Human readable source code is written in an artificial language (such as C, Java, or PHP) that is then compiled (in the case of C) into object code that is then linked to other necessary code before being assembled into a machine readable form known as binary (or machine-code), or is compiled into*

¹² System software usually describes the operating system and related software such as utilities that control and manage the various hardware components and system resources at a level in which they can integrate in operation with one another so that application software may have a well-managed abstraction on which to run when performing specific tasks.

binary bytecode (in the case of Java) that can be executed by a virtual machine (which interprets the bytecode into commands for particular hardware at runtime). Languages such as PHP have their source code interpreted into machine code at the time that the code is executed.

OSS is predicated upon the idea that the human readable *source code* should be available to all. In fact, it is vastly preferred to a binary, *compiled program* within the FLOSS community. This is because the source code can be altered, studied, modified, experimented upon, and thusly improved and extended. Proprietary software has tended to only be distributed as executable binaries, and proprietors attempt to protect their source code from prying eyes so as to protect what is often perceived to be intellectual property. The approach to this access to information is largely what differentiates the “Free” and “Libre” ideologues from their “Open” colleagues. The members of the Free and Libre branches participate in FLOSS due to a professed belief that information should be free (as in liberty) in an open society. The Open branch tends to use the argument that open source development leads to more eyes looking at code which leads to better software. The arguments here are irrelevant to this investigation. The question is to the openness of the source.

Part Two

A Philosophical Investigation of the Concepts of “Open” and “Source” and the Relations to One another In Various Forms of Their Expression in the Presence of Dasein

From Source Code to Sourc(e/ing)

Let us begin by engaging the source. What is the source to which the “open” refers? In OSS, that is clearly intended to be the source code. This should be clear cut, but even at this immediate level of investigation, the phenomenon is elusive. Source code has been described as the human readable code that can be transformed in some way into machine readable, executable code (ultimately, binary). But machine readable code was created by humans in a systematic way, and is thus human readable (even if the process takes training and time to decipher). There is no reason that one could not read Blanchot’s work in translation in binary representations of French (there are reasons that one would not, but these involve similar kinds of reasons that we do not write technical manuals in the linguistic style of Finnegan’s Wake). Furthermore, though there are multiple ways of transforming source code into machine code, the implementation of a language is mostly independent of the language itself (in possibility, if not in theory or practice). C is almost always compiled and PHP interpreted, and they were each intended to be transformed in these respective methods when the languages were designed –but there are C interpreters and PHP compilers. The executable differentiation between binaries and source code is practically irrelevant as a measure of difference in that interpreters can execute code from source in a line by line fashion (though this method is terribly inefficient). So, if source code does not

differentiate itself in its human-readability, or its ability or lack thereof to be executed, how is it that we might know what source code is?

What is it that makes certain code a source rather than something else? The components of the linguistic construction are simple enough, *source* and *code*. The code portion is not the difficulty here. Source code and machine code are each code but of differing statuses. The code as source is what is the differentiating factor. What does it mean for code to be source? This calls for a thinking of source. It also calls for a thinking of *open*. As I said above, there is much at work in this language of *open source*. Let us dance closely with these concepts for a while, following their turns and connections, moving with them as they reveal various intensities and actualities in their emergence as concepts. Source is calling.

As I briefly visited upon the language of source above, I found source to be present in our quotidian language while remaining almost invisible in itself. We are always describing the “source of” phenomena, events, and conditions, the source as such constantly being masked behind the conditions identified as the “source of” things.¹³ We must get into the meaning of source in some way, to find a method of coaxing source out of its articulation in various forms and to identify what source might be as such when it is sourcing.

This language of sourcing is not new. While in contemporary English the word tends to be used in its noun form as a designation of beginning or origin, it has a diverse history in both verb and noun form. In noun form it was used in reference to the sport of hawking and a bird’s rising on the wing, and to refer to the place at which a

¹³ When I refer to *things*, I do not mean objects. Processes, mechanisms and matter all might manifest themselves as things to which I might refer with this term.

flow of water takes its beginning as well as to the quarter whence something of a non-material or abstract character originates.¹⁴ The same word would serve in the coming centuries in verb form as well, each related to their aforementioned noun manifestations—one being that the actual rising of a bird of prey after seizing its quarry was referred to as sourcing, as was the event of rising, surging or springing forth.¹⁵

Source in Relation to Heidegger's Origin

Throughout the common usages and etymology of the term source, there is a common thread. Repeatedly there is a sense of something rising forth. If in common language we speak of a thing's source, we reference its rising forth, but this is still obscure. Would we speak of the "rising forth" of a conflict or the "rising forth of a stream?" We might, but does that say what a source is as such? Something is eluding us, remaining concealed. It seems that while we are closer to what is at work in source, source has yet to be coaxed into the open. Let us return to Heidegger's reference from *The Origin of the Work of Art*.

Origin here means that from which and by which something is what it is. What something is, as it is, we call its essence. The origin of something is the source of its essence.

Heidegger is here treating origin as the source of a thing's being what it is as it is, of its essence. There are two parts to this phrasing, the source of essence. It would be unfortunate to treat origin and source as synonyms as is immediately tempting to do. To do so would lose the significance of source as a subtle differentiation that offers

14 "source, *n.*¹" *The Oxford English Dictionary*, 2nd ed. 1989, [OED Online](http://dictionary.oed.com.proxy.usf.edu/cgi/entry/50231608). Oxford University Press, 5 May 2005 <<http://dictionary.oed.com.proxy.usf.edu/cgi/entry/50231608>>.

15 "source, *v.*¹" *The Oxford English Dictionary*, 2nd ed. 1989, [OED Online](http://dictionary.oed.com.proxy.usf.edu/cgi/entry/50231609). Oxford University Press, 5 May 2005 <<http://dictionary.oed.com.proxy.usf.edu/cgi/entry/50231609>>.

something to us that is not origin. An origin is here articulated as the source of essence. What is the relationship between source and essence that would be called origin? If we walk with Heidegger for a while, and think a thing's essence as its being what it is as it is, then there is a source of this being what it is as it is, and that is what Heidegger calls origin. I will seek to avoid the essence of a thing, and instead investigate source as it can relate to this emergence of the immanent in the space of divergent actualization. I am not immediately looking at how something is as it is, but for how that being emerges into appearance at a source, as the rising forth, the surge.

Source as Expression in Quotidian Language as an Indication of "Catalysis"

When we speak of the source of a river, there is a reference to a rising forth that does not necessarily imply a spring of water rising up from the ground. The source of the Mississippi is the Lake district of Minnesota. Lake Itasca is called its source, though no water springs upward here. Instead, water collects from glacial melt and small streams, each considered too small or remote to be the source of the great river. Lake Itasca is a source in that the Mississippi flows out of it as a river, not yet the mighty Mississippi, but already identifiable as a strong flow with its own emergence as distinct from the previous moment.

When writing a document such as a history or a journalistic article, we reference the source of information, be it a book, the letters of a diplomat, or an interview with a living person. In each of these, we are referencing the source of information. If we are referencing from a work that references another work to get our information, it is called

a secondary source, to differentiate it from the original source. This privileging of the originating source is that it is from whence the information comes, the locus of a claim to the presencing of information. Information is differentiated from data in that information is concentrated and improved data. If we were to record 100 tests of timing the swing of a pendulum, the actual experience of the pendulum swinging would be raw data, but the recording of data in a regular temporal reference such as seconds with measured lengths of the pendulum arms in regular units would be information. Thus, the source of information is that way in which data is recorded (human or otherwise) that is accessible as information. Again there is (*il ya a, es gibt*) some change from one moment to the next, here manifested in the person or location in which data becomes information.

The source of conflict is a difficult case, but one of the most prominent uses of source. To speak of the source of a conflict is to speak of moments charged with the origins of anger, violence, and turmoil. Often cruelty has ensued and communities seek to identify the leading cause or agent that began the conflict. In this articulation, source refers to the point at which violence (physical, emotional, intellectual, etc.) became an option present to the situation.

Another iteration of the language of source is used when addressing artistic inspiration. One may speak of the “source of my inspiration,” or the instant or entity that served to alter my being toward an act of creation. Shelley Jackson’s *Patchwork Girl* utilizes Mary Shelley’s *Frankenstein* as its inspiration. Jackson’s work is a patchwork itself, the text pieced together and interwoven much as the “Modern Monster” her subtitle references. Jackson’s work is original and speaks to its audience in its time. It

transforms the work that served as inspiration for a work and departs drastically in its telling and form. The author when making such a movement from one moment toward an idea can have many reasons for doing so. When beginning the movement due to some sort of experience that might be described as inspired, the source is the the locus of this rapid alteration in state or intensity.

In each of these instances, an alteration is being manifested, some sort of intensity shift that marks a liminal zone, a space (that includes time in its dimensions) in which a limit has been crossed and the emergence of structural/organizational expression is differentiated from processual states in the immediate neighborhood. This process of morphogenesis occurs at an intensified rate or in a compressed dimension. Source is a marking of difference manifested as heterogeneity with the previous moment and the emergence of something newly recognizable. It is not a shift in quality, or intensity, a becoming describable as significantly heterogenous with an immediately previous state, but the local range in which the space of those amorphous processes occur. Manuel DeLanda refers to “anything that switches a dynamical system from one stable state to another” as *catalysis*.¹⁶ In this language, a source would be the local range in which this type of catalysis occurs. Source would be the space shaped by the organizing relationships, intensities, and matter in dynamic complex in which catalysis occurs and from which the consequent state(s) express(es) the difference present in two moments.

16 Manuel DeLanda, “The Geology of Morals,” <<http://www.t0.or.at/DeLanda/geology.htm>>, June 2005.

A Philosophical Description of Source as an Intensive Ordinate through a Return to Heidegger's Origin and an Application of Deleuzian Ontology

The origin of something is the source of its essence.

We return to Heidegger with different eyes. The origin of something is the space that I described above of a thing's essence—this space of a thing's being what it is as it is. But this still does not capture it. This *being* lacks the dynamics of which I have been speaking. For a moment we seemed to be closing upon it, but it still eludes us.

What is the element or addition that will allow this source to source, to emerge as a concept? The quality that is lacking in the reading of Heidegger above is dynamism (this is to say the-reading-of Heidegger-above and not Heidegger's own intention). It is becoming. Isn't this what I have been saying all along? How does this static reading of "being" emerge in a discussion of dynamism, catalysis, and morphogenesis? When thinking these very fluid processes and within this language of difference and transformation, how does this reading of "being" emerge from this thinking the becoming of being? Deleuze presents his thinking on concepts in the opening pages of *What is Philosophy*. In particular, it is his third feature of a concept that speaks immediately to this space in which source is seeking its source.

Each concept will therefore be considered as the point of coincidence, condensation, or accumulation of its own components. The conceptual point constantly traverses its own components, rising and falling within them. In this sense, each component is an *intensive feature*, an intensive ordinate [ordonée intensive], which must be understood not as general or particular but as a pure and simple singularity... In the concept there are only ordinate relationships, not relationships of comprehension or extension, and the

concept's components are neither constants nor variables but pure and simple *variations* ordered according to their neighborhood. They are processual, modular.¹⁷

In this constant traversing of its components, source is seeking to describe the emergence of something in its traversing of its own coincidence in the moments of the accumulation of this coincidence. Source is an intensive ordinate of the difference between two moments spanned by the action of catalysis. Its being is becoming as much as the becoming it seeks to describe, but the becoming of source is in its own variations and not in those variations of the spaces which it describes as a sourcing. Source as a concept has its own range. This is what I have been seeking to traverse, to accumulate in my understanding, in order that it may give itself as this processual thing. Something is slipping. There is a condensation at work in this traversal. Return. Another iteration. Each repetition expressing difference. Each iteration opening difference as the revealing of source as what it is becoming as it becomes.

The origin of something is the source of its essence.

The origin of something is the source of its essence. The source of some thing's essence is its origin. The source of a thing is the space in which a thing emerges through the traversal and accumulation of its component relationships, intensities, and actualities that mark the heterogeneity of this complex in two moments spanned by a catalysis. The origin of something is the space and components which shape that space in which a thing reveals itself as a becoming of what it is becoming in that space.

¹⁷ Gilles Deleuze and Felix Guattari, *What is Philosophy*, trans. Hugh Tomlinson and Graham Burchell, (New York: Columbia University, 1994), 20.

Source is the space—shaped by a thing’s own emergence—in which the thing reveals its becoming.

Put another way: source is the space of a thing’s disconcealing in time, the space of an entity’s individuating process.

Source as the Space of Emergence into the Open of Dasein’s Being-Open-To Her Disinhibitor in the Absence of (Dis)Inhibition

This sourcing through traversal, accumulation, coincidence, et al involves processes and relationships that are each open to others. This openness being the receptivity of components to form relations, to draw from and into one another. But this open is not open to us yet, it needs opening. In his *Parmenides* lectures, Heidegger writes of this type of coming forth and emerging into appearance, into the *open*.

This reflection will seem strange to the ordinary view, especially because it shows that the open is by no means first and only a result or consequence of disclosure but is itself the ground and the essential beginning of unconcealedness. For, to disclose, i.e., to let appear in the open, can only be accomplished by what gives in advance this open and thus is in itself self-opening and thereby is essentially open, or as we may also say, is of itself already “free.” The still concealed essence of the open as the primordial self-opening is “freedom.”¹⁸

In this, source and the open are drawn together. The moment on the near side of the catalysis of the space of the source is the open. I say near side, because a thing sourced has emerged on this side of the fold, the side of the world, our world. It has revealed itself, but revealed itself to whom? To whom can a thing reveal itself through its sourcing? When a thing enters the open, what is open to it?

¹⁸ Martin Heidegger, *Parmenides*, trans. André Schuwer and Richard Rojcewicz, (Bloomington: Indiana University, 1992), 142.

Giorgio Agamben has devoted an entire work to the open. In this work, he explores differences between humans and their fellow organisms. All such beings have *disinhibitors*, those things in an environment which elicit a response, or affect that organism. Behavior is elicited through the recognition of the environmental stimuli that disinhibit a being from its existing state, enabling it to respond. Agamben differentiates—with Heidegger—between the animal's being open to its disinhibitor without being open to the being of this disinhibitor *as such* and the human as Dasein opening to a disinhibitor as the thing in its Being.¹⁹ This is to what Heidegger refers when he speaks of the animal's poverty in world. While this distinction between animal and human is too clear (and I think Agamben would agree), we can still speak of this thing that seems peculiar to few species on our planet in varying degrees, if not only humans, this being open to the being of the disinhibitors of behavior in a way that is not captivation and includes an awareness of captivation in its lack. It is this sort of disconcealing process with the disinhibitors of human behavior that allow for rich formation/alteration of human worlds.

Returning to source, the earth is always already at work, much of it concealed from appearance, variations in intensity generating processes of transformation. The local individuating process of a thing occurs in a space in which a thing emerges into the open of appearances where humans can open to these disinhibitions of their behavior. How do humans open themselves in this way to the disinhibitor in a relationship that differentiates it from that which captivates most organisms? How can they and any like them enter into relation with the realm of the open and its freedom?

19 Giorgio Agamben, *The Open*, trans. Kevin Attell, (Stanford: Stanford University, 2004), 51-53.

Agamben explores Heidegger's thinking on this and comes to the source of the Dasein in profound boredom, in the suspension of the animal relationship with a disinhibitor. In this lack of response, the Dasein is bored with the stimulus of its behavior. It does not react and is thus exposed to the thing that would affect it but does not, and it is in this indifferent relation that the human might open to the thing as a thing and not simply to the thing as stimulant of behavior.

From the beginning, being is traversed by the nothing, the *Lichtung* is also originarily *Nichtung*, because the world has become open for man only through the interruption and nihilation of the living being's relationship with its disinhibitor. To be sure, just as the living being does not know being, neither does it know the nothing; but being appears in the "clear light of the nothing" only because man, in the experience of profound boredom has risked himself in the suspension of his relationship with the environment as a living being... Dasein is simply an animal that has learned to become bored; it has awakened *from* its own captivation *to* its own captivation. This awakening of the living being to its own being-captivated, this anxious and resolute opening to a not-open, is the human.²⁰

It is Dasein which is open to a thing's opening itself in its sourcing. This is an opening at a level of abstraction, as a virtuality, an intensive difference that Dasein infers from singular accidental phenomena that reveal themselves as concealed in the folding of their unfolding. Things may be open to one another in the sense that they are able to affect one another, that they may enter into relation and alter one another, from particles to parenting. It is Dasein's experience of a thing as a thing—removed from it—and thus an experience of Dasein itself in relation to its indifferent object that is peculiar about this relationship amongst Dasein, its disinhibitor, and its captivation. The space of this relationship is the open. It is the space in which *alétheia* can happen as the moment of truth.

²⁰ Ibid., 69-70.

But for us now it is a question of thinking the essence of freedom in essential unity with the most primordial concept of *alétheia*, and indeed with a view to elucidating the essence of the open. Thereby we might grasp the freedom man must first attain, in accord with his essence, if he is to be able to let beings be in the open what they are as beings... The free is the guarantee, the sheltering place, for the Being of beings... We ordinarily think of the open, the free, and the vast as conditions of scattering, dispersion, and distraction. The open and its extension into the vastness of the unlimited and limitless are zones without stopping places, where every sojourn loses itself in instability. The open provides no shelter or security. The open is rather the place where what is still undetermined and unresolved plays out, and therefore it is an occasion for erring and going astray.²¹

In this small block of text, Heidegger touches upon much of what generates anxiety in the open. Let us open to open a bit more in order to understand the array of human methods of being toward the concept of open.

Open To, Open(ness), and the Open as Exposure and Appearance in the Presence of Dasein

There are more than twenty pages devoted to the word “open” and its variants in the Oxford English Dictionary. The words that surround this concept are many: gap, space, unconcealed, plainly seen, in public notice or view, unenclosed, without cover, opportunity, without obfuscation, free from obstruction, access or passage, affording unrestricted access or entry, bare, exposed, revealed, vulnerable, not finished or completed, disclose, available, to spread out, expand, unfold.

In thinking through source, we explored Deleuze’s thinking on a thing’s components, coincidence, and accumulation. There is a technique here, a method for investigating a phenomenon in order to enter into a being-with its concept. We might enter into relation with a thing’s components in order to witness what is revealed in its

²¹ Heidegger, *Parmenides*, 143-4.

traversal through and its accumulation of the coincidences of these components. What is at work within these various concepts as relate to *open*—the relations amongst and the intensities common to them? Gaps and spaces imply some lack of bordering, a break in a boundary, a discontinuity, some access or passage which reveals or grants access. This lack of obstruction or obfuscation granting opportunity or revealing vulnerability, laying bare what is there, spreading it out, unfolding what is hidden in the folds. Public notice or view implies a general access, unfiltered. The language of vulnerability and opportunity are related but object driven, requiring a judgment to differentiate them. The concept *exposed* is within each. There is a missing object here, to what is a thing exposed, there is a “to” lacking articulation here. Exposed *to*. Exposed through a gap, some porous boundary allowing an unfiltered transfer across its border of material, energy, or information. This is open, but the open is too open for this to encapsulate it.

There seem to be three senses of open at work here and this may be confusing our investigation. First, there is the sense of open as the act of opening to. Second, there is the quality of being in a state of openness. And third, there is the open in which things may emerge. The three are related but by not teasing out the expressions of the relationships, there is a confusion that does not contribute to the concepts’ unfolding, to the ability to find the virtual at work that might open each to our understanding.

The first sense of opening to is closely related to the inquiry concerning source. When a thing opens to another being, there is an extension of possible interaction. The thing opened has provided a gap, an interface to the space outside its space of becoming. It has opened its steady state to perturbation. This extension of component influences is the opportunity of change through the attraction, repulsion, interaction

and infection of other spaces. When a thing is sourced and reveals itself as itself is such an instance. It is opened to Dasein and her world's processes, relations, and intensive differences—her worlding. The second sense of open as the state of being is the state of exposure that lies across the catalysis of an opening to. This is not to say that the state of open(ness) is a static state, but that it is in state of being open until it is closed in varying degrees or to miscellaneous spaces of other beings. A thing is in the state of open(ness) until it conceals itself again in its (un)folding in time, at which point the earth reclaims the actuality of the virtual and the space of the the thing's coherence as a thing dissipates or implodes. This brings us to the third expression of open as the open of appearance.

When I spoke of a thing's being open to the miscellaneous spaces of other beings, what I was seeking to imply is the existence of partial spaces, component spaces that bleed into one another with singularities and attractors. The gravities of intensive differences shape each space and all space as interweaving meshes of influence, attraction, repulsion, and transfer continuously altering state—states ranging from precarious homeostasis to somewhat stable emergent structures coming into being through morphogenetic processes driven by these intensive differences. Each partial space has its own degrees and ranges of open(ness). The widest range in which all of these exist as the most expansive space open to Dasein's experience thereof is *the* open. The open is the lived ground of disconcealment into the world—with Dasein open to the appearance of things in their open(ness) in the open. The passage from the undisconcealed earth to alétheia in the open of the world is the singular event of the source. I have returned to the source through the opening of the open. From the

investigation thus far, it becomes apparent that the concepts of source and open are in relation to one another. But how are they in relation?

Recapitulation

- i. The passage from the un-disconcealed to the open ground of *alétheia* is source.
- ii. Source is the space—shaped by a thing's own emergence—in which the thing reveals its becoming—it is the space of a thing's disconcealing in time, the space of an entity's individuating process.
- iii. I explored three faces of open:
 - a) open as the event of opening to
 - b) the state of being open
 - c) the open as ground for world
- iv. The event of opening (a) into the open(ness) (b) of the open (c) is the sourcing of the appearance of the thing to Dasein as disconcealing.

In this thinking, the open is the space of appearance of the phenomenon. The phenomenon is a thing experienced as such, which is both the thing's existence as actuality and the expression of its intensive difference engine as the virtual. The emergence of this somewhat steady state of actual-virtual in the worlding of the real occurs via the singularity of its source. In this, the source is the singular space of the immanent virtuality of the phenomenon which opens to the open of Dasein's boredom with its disinhibitor. The source marks the phenomenon's emergence into the open, and thus the entry into the world of human life. The open is the space of appearance into which the singularity of source opens, and is thus the ground of the world and the unfolding of phenomena.

How a Source Becomes Open Through an Opening of its Opening To the Intensive Differences of an Inhabited Open

This brings us back again to open source. If we can speak of open and of source and of their relations as virtual entities, can we speak of a source being open rather than merely sourcing into the open or a source opening to the open? How might a source be open? This would be an example of the second state of open described above.

The second sense of open as a state of being is the state of exposure that lies across the catalysis of an opening to. This is not to say that the state of open (ness) is a static state, but that it is in state of being open until it is closed in varying degrees or to miscellaneous spaces of other beings. A thing is in the state of open(ness) until it conceals itself again in its (un)folding in time, at which point the earth reclaims the actuality of the virtual and the space of the the thing's coherence as a thing dissipates or implodes.

An open source would be a source that is opening unto the open. But source already does this, source is in some sense always open in that it is a disconcealing, a making actual of virtualities in the open space of appearance as phenomenon in the presence of Dasein's openness to the concept of the phenomenon. What differentiates an *open source* from a *source*? What is the intensive difference? In thinking source as the intensive ordinate of a things actualization, the open has always been present. Perhaps the difference lies in the space of this emergence at the moment of sourcing, in the range of the thing's opening to the open. The open involves a being exposed.

The concept *exposed* is within each. There is a missing object here, to what is a thing exposed, there is a "to" lacking articulation here. Exposed *to*. Exposed through a gap, some porous boundary allowing an unfiltered transfer across its border of material, energy, or information.

If a source can be open to more than one combinatorial, interactive, perceptive possibility, then there may be degrees of complexity to a source's opening. A source is not simply opening to the open, but opening to the open of an inhabited space—a space

inhabited by other difference engines and partial spaces. Perhaps this surplus of openness in the designation of *open* source is an indicator of the degrees of partial spaces to which the source opens. These partial spaces to which a source might open, is not an ordinary particular, but the singular particular of the Dasein, as it is the open of Dasein's openness to the phenomenon as a phenomenon that distinguishes the space of the open. Thus the greatest range of open to the source would be to open not to the greatest number of spaces as Dasein as it might be tempting to think, but to be open to the open of Dasein spaces without number. But there is still something not giving itself in this description. This is still an opening into the open of appearance that has been described from the beginning.

When thinking of open above, I had said:

When a thing opens to another being, there is an extension of possible interaction. The thing opened has provided a gap, an interface to the space outside its space of becoming. It has opened to a new horizon. This extension of component influences is the opportunity of change through the attraction, repulsion, interaction and infection of other spaces. This occurs when a thing is sourced and reveals itself as itself.

In becoming focused upon the disconcealing of a thing to the open(ness) of Dasein, my vision lost its periphery. This *extension of possible interaction* is not simply the appearance, it is emergent combinatorial/interactive interfaces as well. If a thing is only open to appearance (and it never is), then it has only one degree of freedom, one degree of movement in the space of the open. If a source is open in the sense of being open in its opening—opening to transformation in its act of opening to the open—then we might call the source “open” to distinguish it from the event of sourcing into the open of appearance as phenomenon.

To what might a source extend its possible interactions when opening to the open as open? The source is opening to the open of appearances, the actualization of a concept. Thus any thing capable of interacting with the things sourced as concept would be in the range of this extension. To what a thing is open would be the domain of things to which the range of openness might be extended. For source, this is Dasein. To open a source to transformation through intensive differences with Dasein would be to open the source. An open source would be a source in the state of being exposed to whatever intensive differences exist between beings actualizing the concept of Dasein and that source.

Open source is a space—shaped by a thing's own emergence—in which the thing reveals its becoming to Dasein and remains exposed to whatever intensive differences exist between all Dasein and that space of a thing's immanent emergence as phenomenon with Dasein.

Part Three

A Re-Examination of Source Code, Its Differentiation as Open Source Code, and an Analysis of The Open Source Definition in the Light of The Previous Philosophical Investigation

It has been a long road since we began this investigation of software movements and programming languages, compilers, licenses and hackers. I embarked upon this process through the posing of a problem: *What is source code and how can it be open?* Together we have moved through this thinking, repeatedly tapping the phenomena to hear its report. Recording the movement in order to return with difference. It is time now to return to the phenomena with which we began. We left many questions to which we now return:

1. What is it that makes certain code a source rather than something else?
2. What differentiates source code from open source code?
3. What is described in the phrase "Open Source Software?"

I have already argued that the distinction between source code and machine code as human readable versus machine readable is flawed. But returning to the question after this long sojourn, it seems that the human readable distinction may not be far from the mark. Perhaps this is an example of seeking significance in a steady state as masking the dynamism in which we might find the phenomenon more giving.

Code is not simply an artifact, it emerges. It is a work, in some ways similar to a work of art (perhaps more than is immediately apparent to those uninitiated to computer languages). It lies just this side of the coder. The coder becomes coder by virtue of participating in the production of code. Code and coder become together for at

least the moments of writing. If we look back an instant before the moments of coding, the code did not exist. Code enters the space of the open in the event of coding. The component parts: language, the space of the Dasein with all its constituent spaces in relation, artificial language, programming models, the machine, keyboard, and monitor all coming together in a moment in which none of them is code and code is not apparent. Catalysis happens. There is a morphogenetic dynamism at work. There is a sourcing here.

In this constant traversing of its components, source is seeking to describe the emergence of something in its traversing of its own coincidence in the moments of the accumulation of this coincidence. Source is an intensive ordinate of the difference between two moments spanned by the action of catalysis.

The code that emerges in these moments of actualization is source code. It might be said that when any code passes through an interpreter or compiler, that this is a sourcing of code as well, but this is not the case. We spoke of sourcing as emerging into the realm of the open. When code is compiled or interpreted it is not sourcing, it is being compiled or interpreted. These are translation mechanisms to various degrees with compilation including a layer of gathering and linking. The code is not being sourced, it is already in the open. *Source is the space—shaped by a thing's own emergence—in which the thing reveals its becoming.* The revelation has occurred, there is a transformation at work in translation, a catalysis that will take the source code and alter its state, but this process adds nothing to the code's becoming that actualizes the virtual in the presence of Dasein's open. Source code is the code as it has emerged from the virtuality of a software model or idea without actuality as it is written on particular machines at particular times by particular programmers in particular moods with particular target platforms in mind. It emerges

from a generative process with at least two actualizations, coder and code. Source code is the artifact of that sourcing—but not simply.

Open source is a space—shaped by a thing's own emergence—in which the thing reveals its becoming to Dasein and remains exposed to whatever intensive differences exist between Dasein and that space of a thing's immanent emergence as phenomenon with Dasein.

How would source code actualize something like this? A source code, as presented above has already established that it has opened itself as code in the moment of coding. It has shaped its space in its coming into being and found a steady state. What has not been established is how source code “remains exposed to whatever intensive differences... exist between Dasein and that space of a thing's immanent emergence as phenomenon with Dasein.” This *remaining exposed* of the space of the source is the rub. If source code has its own space in its sourcing into the open of Dasein, then what this is describing is the source's space being held exposed to open and being opened itself.

If a source is open in the sense of being open in its opening—opening to transformation in its act of opening to the open—then we might call the source “open” to distinguish it from the event of sourcing into the open of appearance as phenomenon.

The source code is not simply being sourced into the open, if it is open, it is opening the source code itself to transformation as well. It is locking open the space of the source, but not only as a window to a singular point in time, but as continuous transformation and emergence. The source code is from-this-point-onward exposed not only to appearance, but to transformation of its constant co-emergence with coder-Daseins returning to the source with difference.

OSS is software that is freely distribut(ed/able) in both source code and

compiled form. Any software derived from software governed by an Open Source license must be governed by the same license as the original software. The distributed software should not be restricted to any person, community, software environment, etc. or in any way or be obfuscated so as to de facto restrict the readability of the source code without providing readily available alternatives that would render the software to a state as if it had none of these restrictions.

This earlier description of Open Source Software derived from the licensing guidelines from the Open Source Initiative brings me back to here. It is remarkable how closely it follows the investigation that has been building in the movement throughout this paper. The software is freely distributable in both source *and* compiled form, while the examination above only required that the source remain open to readability by other humans. The derivative products structure holds the source code open in time. As the code is transferred altered, etc., it holds the source open to reading, modification, and distribution. It expands the number of Dasein participants to the number that choose to enter into the open with the source code and forbids restriction of access to this space. It also requires that any alterations made to the source code that violate the openness of the space of the source be reversible via a readily available and simple process. In all of these, the Open Source Definition seems to actualize in its license and the code that actualizes it the concept of Open Source as it has emerged here. This could be done without a license, as a guerilla act, but the Open Source Software licensing model creates a practical legal construction that protects the space of code's source.

Life Techniques that Overcome Ourselves

While the impetus for this investigation emerged from my exposure to the language of Open Source Software, the investigation provides us with a sense of

understanding open sources as a way of being, a technique for living in the media life of which Schirmacher has written and taught. It is not a tool, in that equipment is exhausted in its utility and closes Dasein's being into the box of usefulness. Living as open source is to move toward the life that Sartre imagined. To do so would be the actualization that would render the ethical argument moot, and the human overcome.

Reluctantly

that which dwells near its origin abandons the site.

--Holderlin, "The Journey," versus 18-19²²

²² Heidegger, "The Origin of the Work of Art," 203.

Addendum

Manuel DeLanda states in the talk “Open Source: A Movement in Search of a Philosophy,” a paper that he delivered to the Princeton Center for Advanced Study in 2001, that while the Open Source movement lacked a robust philosophical foundation, this was irrelevant to the quality of the software produced by the movement. He goes on to say that he would not attempt to create such a philosophy—though he does provide a rather interesting economic model for the grounding for the movement—but that if such a philosophy were to evolve, it should “follow a similar path of the software, that is, be the collective product of the users of that philosophy.”²³

With this in mind, I have decided to open this work as the originating source for the development of an open source philosophy. DeLanda wrote in 2001 that he believed it to be premature for such a philosophy. I believe that it is untimely enough now for this to begin to emerge. The website <http://opensourcephilosophy.org> has been reserved and I will begin to transfer this document into it in parts that will be open to study, modification, distribution by anyone under the guidelines of the GNU Free Documentation License. This is not about “freedom” or “good” philosophy, though I hope that this gesture will actualize these, it is about actualizing this work as a hack, creatively overcoming or circumventing limitations—in this case, my own.

²³ Manuel DeLanda, “Open Source: A Movement in Search of a Philosophy,” <<http://www.cddc.vt.edu/host/delanda/pages/opensource.htm>>, June, 2005.

Appendices

Appendix A

The Open Source Definition

Version 1.9

*The indented, italicized sections below appear as annotations to the Open Source Definition (OSD) and are **not** a part of the OSD. A plain version of the OSD without annotations can be found [here](#).*

Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

***Rationale:** By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect.*

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

***Rationale:** We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.*

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

***Rationale:** The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.*

4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of “patch files” with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

***Rationale:** Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.*

*Accordingly, an open-source license **must** guarantee that source be readily available, but **may** require that it be distributed as pristine base sources plus patches. In this way, “unofficial” changes can be made available but readily distinguished from the base source.*

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

***Rationale:** In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.*

Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

***Rationale:** The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.*

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

***Rationale:** This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.*

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

Rationale: This clause forecloses yet another class of license traps.

9. The License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

Rationale: Distributors of open-source software have the right to make their own choices about their own software.

Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.²⁴

²⁴ Open Source Initiative, "The Open Source Definition,"
<<http://web.archive.org/web/20011019055512/opensource.org/docs/definition.html>>, July, 2005.

Appendix B

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

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

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) 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 this service 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 make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. 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.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it

clear that any patent must be licensed for everyone's free use or not licensed at all.

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

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute 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 and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program

itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), 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 distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the 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 a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, 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.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE 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.

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 convey 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 an idea of what it does.
Copyright (C) yyyy 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 2 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, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

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

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

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision 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, the commands you use may be called something other than `show w’ and `show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This 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.²⁵

²⁵ Free Software Foundation, “GNU General Public License,” <<http://www.gnu.org/copyleft/gpl.html>>, July, 2005.

Bibliography

- Agamben, Giorgio. *The Open*. Translated by Kevin Attell. Stanford: Stanford University, 2004.
- DeLanda, Manuel. "Deleuze and the Open Ended Becoming of the World." Available at <http://www.diss.sense.uni-konstanz.de/virtualitaet/DeLanda.html> Accessed April 2005.
- DeLanda, Manuel. "The Geology of Morals." Available at <http://www.t0.or.at/DeLanda/geology.htm>. Accessed June 2005.
- DeLanda, Manuel. "Meshworks, Hierarchies and Interfaces." Available from <http://www.t0.or.at/delanda/meshwork.htm>. Accessed April 2005.
- DeLanda, Manuel. "Open Source: A Movement in Search of a Philosophy." Available at <http://www.cddc.vt.edu/host/delanda/pages/opensource.htm>. Accessed June 2005.
- Deleuze, Gilles. *Difference and Repetition*. Translated by Paul Patton. New York: Columbia University, 1994.
- Deleuze, Gilles and Guattari, Félix. *A Thousand Plateaus*. Translated by Brian Massumi. Minneapolis: University of Minnesota, 1987.
- Deleuze, Gilles and Guattari, Felix. *What is Philosophy*. Translated by Hugh Tomlinson and Graham Burchell. New York: Columbia University, 1994.
- Heidegger, Martin. *Being and Time*. Translated by Joan Stambaugh. Albany: State University Press of New York, 1996.
- Heidegger, Martin "The Origin of the Work of Art," In *Martin Heidegger: Basic Writings*. Edited by David Farrell Krell. New York: HarperCollins, 1993.
- Heidegger, Martin. *Parmenides*. Translated by André Schuwer and Richard Rojcewicz. Bloomington: Indiana University, 1992.
- Nietzsche, Friedrich. "Twilight of the Idols." In *The Portable Nietzsche*, Edited and Translated by Walter Kauffman. New York: Viking Penguin, 1976.
- Open Source Initiative. "The Open Source Definition." Available at <http://web.archive.org/web/20011019055512/opensource.org/docs/definition.html>. Accessed July 2005.

The Open Source Initiative. "Site Index." Available at <http://www.opensource.org/index.php> . Accessed August 2005.

The Oxford English Dictionary Online. Oxford University Press. 5 May 2005
<<http://dictionary.oed.com.proxy.usf.edu/cgi/entry/50231608>>.

Raymond, Eric S. *The Jargon File*. Available at <http://www.catb.org/~esr/jargon/html/H/hacker.html>. Accessed July 2005.

Schirmacher, Wolfgang. "Eco-Sophia: The Artist of Life." Available at http://www.egs.edu/Art_Life/wolfgang/ecoso.html. Accessed October 2004.

Schirmacher, Wolfgang. "From the Phenomenon to the Event of Technology." Available at http://www.egs.usf.edu/Art_life/wolfgang/event.html . Accessed September 2004.

Schirmacher, Wolfgang. "Homo Generator: Militant Media and Postmodern Technology." Available at http://www.egs.edu/Art_Life/wolfgang/homo.html . Accessed September 2004.

Schirmacher, Wolfgang. "Media Aesthetics in Europe." Available at http://www.egs.edu/Art_Life/wolfgang/aesth.html . Accessed October 2004.

Schirmacher, Wolfgang . "Privacy as an Ethical Problem in the Computer Society." Translated by Virginia Cutrufelli. Available at <http://www.egs.edu/faculty/schirmacher/privacy.html>. Accessed June 2005.

Whitman, Walt. "Song of the Open Road." Ed. Francis Murphy. New York, 1996.