Ethics and Information Technology (2004) 6: 223–231
DOI 10.1007/s10676-005-0350-7

Digital nominalism. Notes on the ethics of information society in view of the ontology of the digital

Tere Vadén
Department of mathematics, statistics and philosophy, University of Tampere, 33014, Finland
E-mail: tere.vaden@uta.fi

Abstract. The commodification of code demands two preconditions: a belief if the existence of code and a system of ownership for the code. An examination of these preconditions is helpful for resisting the further widening of digital divides. The ontological belief in the relatively independent existence of code is dependent on our understanding of what the "digital" is. Here it is claimed that the digital is not a natural kind, but a concept that is relative to our practices of interpretation. An interpretative system that sees code as something that can or should always be owned implies an increase of social control and threatens vital processes of knowledge creation that are necessary for an open and egalitarian information society. The ontological belief in "digital code" thus provides the backdrop for an ethical view of the information society. Consequently, if we see digital code as an interpretative notion (in the nominalist way), the ethical questions appear in a different light.

Key words: digitalness, digital divide, disorganisations, free software, ICT, information society, intellectual property, knowledge creation, ontology

Introduction: The digital information society, code and postmodern philosophy

The concept of the digital contains a plethora of hopes and fears. A new way of "digital being" is thought to prompt both the need for new digital ontologies and epistemologies, as well as to result in novel ethical dilemmas. These expectations have a real effect when the so-called information societies are given shape by concepts like "intellectual property" and "digital rights management". If the central feature of the information societies is the role of digital information as a tool and as a product, then the definitions and interpretations of terms like "information", "knowledge" and the "digital" are of essential importance. Crucially, digital information allows for new configurations of basic social forms like property and knowledge. Because digital information can be reliably and cheaply copied, it has great potential for democratisation and equality of information distribution. On the other hand, digital technology also makes possible the monitoring and controlling of information so that the commodification of knowledge is possible in ways that threaten to create huge asymmetries or so-called digital divides.

The ontological idea of the "being of the digital" or of "digital being" is closely connected to the idea of the immateriality and pervasiveness of information. This idea of disembodied information, immaterial knowledge and artificial intelligence (AI) is a commanding tenet in the tradition that sees a utopia in "digitalness". The utopia of AI as well as the utopia of digital being are based on the idea that some processes or entities in the world (digital, algorithmic, informational, intelligent, cognitive processes; in a word "coded" processes or entities) are defined and given existence by their abstract forms in a way that makes the details of their physical constitution irrelevant.[1] The coded processes are multiply realisable in different physical make-ups. The utopia rests on separating two different ontological and epistemological areas. On the one hand we have the abstract and functional world of digital code (information, representations), on the other hand the concrete and obstinate corporeal world of physical instantiations. These two worlds, the experienced physical world and the idealised abstract world, are thought to be isomorphic, and this isomorphism is though to be the basis of our knowledge of the abstract world.[2]

Along with the ontological and epistemological questions, digital information processing raises

[1] As a level of explanation this coded existence is called functionalistic: one discusses the functional roles of abstract entities in separation of their physical realisation; see, e.g. Block (1980).

[2] The idea of isomorphism goes back at least as far as Plato.

ethical questions on the global scale, not the least because of commodification of code (intellectual property rights, information patents, patents on DNA, copyrights on immaterial assets etc.; for a review of current legal and social developments, see Drahos and Braithwaite 2002). The commodification of code and, consequently, of knowledge embedded in code needs two preconditions. For the first, what is needed is a sustained belief in the existence of "code" in which information content can be expressed, transferred and through which it can be identified. This is the ontological side of the issue: a belief in code as a more or less independently existing entity. For the second, one has to set up a system of ownership for this code. This is the socio-political, or, more generally, ethical side of the issue. In this sense, the first condition does not only comprise of a scientific way of looking at the world and explaining phenomena as if they were in the last instance algorithmic or "coded". It also provides the necessary background for a particular view of the ethical implications of information society development. To put the matter in a crude simplification: if code did not exist, it could not be owned. Consequently, different views and interpretations on exactly how and where code does exist imply different views on the ethics of information societies. "Code" is a both scientific term (a theoretical term, if you wish) with an epistemology and ontology of its own, but also a governmental and juridical concept invested with economic and political interests. It is becoming clear that the first condition is closely coupled to the second condition in a world where economic production is dependent on intangibles. For example, "intellectual property" as a concept is a way of controlling and organising – in the Taylorist sense – the new type of production.[3]

If it indeed is the case that the notion of "intellectual property" functions not only as a basis for a kind of information society but also produces effects that widen the digital divides between owners and non-owners of information and knowledge, it is of essence to note that the processes and the concepts behind them – "code", "digital", "property" – are historically and socially contingent. The development of information societies is in part dependent on how these concepts are understood and employed.[4] One of the most debated and profound themes in 20th century philosophy was the controversy over the nature of technology. On one extreme, technology was seen as a neutral, harmless and helpful tool for humankind to use, on the other extreme, it was seen as a pervasive power out of human control and with pernicious goals of its own. Marxist thinking, Popperian rationalism, critical theory, and Heideggerianism all have their own distinct views on the matter. The questions of the non-neutrality of technology, in general, and of the interests included in the use of a particular technology become salient if we continue to take the question further by asking, in a postmodern vein, what structures of power are embedded in these interests, what productive effects (in the Foucaultian sense) they have, or, more bluntly, who or what groups of people benefit from a particular use or form of technology. The interesting questions about technology, including digital information technology, can only be formed when this kind of "who"-analysis (analysis of the structures of power) is combined with the insight that technology is not one thing, not one identifiable whole with a lasting essence or drive leading to particular kinds of society. "The same" kind of technology (its forms and use) may now benefit this group of people, the next day that group of people or form of life. So the questions have to be more concrete: what forms of power does a particular way of using technology support, need, presuppose or undermine? What are the forms of subjectivity or community included in a particular understanding and use of information technology? What kind of technology does a particular social regime support? These questions are available only if both society and technology are seen as non-essentialistic entities, with no immutable or everlasting features.

Through this set of questions technology loses its neutrality in both of the senses: it cannot be seen as a pure tool or as something that forces something on us. Naturally, technology is also a tool. However, understanding a particular technology as a tool has in each instance specific consequences. Equally, technology as a way of thinking is beyond our will as persons. But these two extreme positions only make the questions more pressing. These questions link the uses of technology into the manifold and contradictory social, cultural and political developments and power structures that develop also for reasons relatively independent of technology. These power

[3] Copyright legislation and the laws concerning other so-called immaterial rights are being changed at a quick pace. This trend is most marked in the US, but the EU is following close behind. It has been claimed that "never before have so few owned so much a part of intellectual property" (Lessig 2002, see also Lessig 2001); these words do, intentionally or not, have a familiar ring.

[4] For an interesting historical account on how technology influences our relations to documents in the general sense of the word, see Levy (2001).

struggles and technological interests are visible, for instance, in the practices organised by the concepts of "code", "digital" and "property".

Code and its existence

Two ways of resisting the commodification of code

The hypothesis is, then, that what was called "commodification of code" above is a combination of two information technological interests: (1) the belief in a functioning layer of code (to be separated from the physical instantiation) and (2) the set-up of a system of ownership for the code. If this is the case, two ways of counteracting the development of an increasingly asymmetric information society present themselves. One straightforward stance is to accept the existence of code (to believe in it, and act accordingly), to defend it as a way of understanding the world, but at the same time insist that the ownership of code leads to blatant contradictions, as well as to social injustice and economic imbalance. One of the ensuing contradictions is maybe most evident in the case of scientific knowledge (including theories represented in formal code, such as mathematics, large parts of natural science, programming, etc.), which receives its special status and credibility from the very fact that it is not owned: knowledge becomes scientific only through the open and free critique of the scientific community. To quote Jacques Derrida: "[...] in a scientific text [...] the value of the utterance is separated, or cuts itself off, from the name of the author without essential risk, and, indeed, must be able to do so in order to lay claim to objectivity." (2002, p. 47). As a speech act, a scientific text has to be distinguished from the person or persons who "sign(s)" it, otherwise we are not dealing with a text that can assume the special characteristics and authority of a scientific text. This has been and still is largely the way in which scientific information and knowledge are severed from a concept of private property that is dependent on the link between a person and an entity. The author, the one who "signs" science, is the scientific community. A particular way of speaking, a particular type of speech act, namely that of scientific texts, creates a community and a way of appropriating knowledge that is different from the case of private property (as understood, e.g., in the Lockean sense).

A similar device for co-operating without the intrusion of private property has been developed in the case of computer software. So-called free software is developed by a community of people who share-and-share-alike: the goal is to develop software that the user is free to use, modify and redistribute provided that the freedoms are transferred. In this sense the ideal is close to the ideal of science. For this purpose the movement needs a legal and social tool, one that uses the copyright claim set on a piece of software for community building rather than private property building. This tool, developed by Richard M. Stallman, is often colloquially called "copyleft": the copyright statement in question gives the user the right to modify and redistribute (the modified version of) the software provided that the right is also transferred (see Stallman 2002). This viral nature of the "copyleft" copyright protects the information and knowledge amassed in the software from becoming closed by ownership. The knowledge is appropriated inside the common control of the community.

In both cases, science and free software, the condition of the practice, as well as its goal, is a community of sharing based on a certain set of common values and interests. Both can be seen as ways of acting, as power structures, that are instrumental in creating an information society that contradicts the trends of commodification of code and knowledge. As such the practices of these communities also demonstrate that digital information processing (or any other technology) does not force us treat code as property.

The second way of resisting the development of an increasingly asymmetric information society is to direct our attention to the first condition, to the process of seeing the most salient features of the world as code and, especially, to the peculiar kind of existence that code has. This questioning can be set apart from the questions concerning ownership, but my aim in the following is to connect the questions of the existence of code to problems of the social and cultural status of code. Thus the ontological problems are intended to guide the view on the ethical problems of information societies.

Digital existence

How does "the digital" exist? The digital is not a precisely defined technical, philosophical or scientific concept. It is also important to emphasise that the digital is not a natural kind: it is impossible to build a "digitalness-detector" that would start ticking whenever something digital is present. The devices called digital are so dissimilar in their structure, physical construction and functions that distinguishing them from other types of devices is very hard. Also, the distinction between the digital and non-digital functioning of a digital device is a tenuous one. For instance, considered as a physical entity, a digital

computer functions essentially according to the same principles when it does what it is supposed to do and when it doesn't (due, say, to a broken wire). The word "digital" is essentially an engineering concept that has to do with the way in which electronic devices are supposed to work. To be more precise, "digitalness" is a demand on the reliability of the functioning of a device.

One way of approaching the particular type of existence that the digital has is to look at the opposite of the digital, the analog. A quick and coarse definition is to say that analog devices work on continuous scales and are sensitive to the whole scale of a relevant factor, while digital devices contain discrete jumps from one state to another. The Latin "digit", toes and fingers, includes this possibility for jumpy and bitwise operation. One can count by using ones fingers to represent discrete numbers, jumping from 3 to 4 without going through the continuous scale from 3 to 4. Letters are digital in the same way. Where one letter ends another starts. A definition of the exact point of transition is not necessary as long as the letters can be separated and recognised reliably enough.

The efficiency and alluring beauty of digitalness is a direct follow-up of this discreteness and step-by-step interpretation. At least in principle it is possible to repeat a process of a digital device at will in exactly the same way (for example, the transition from 3 to 4). To be more precise, it is possible to repeat a process so that the features considered relevant are preserved. In this way it is possible to copy, store and reproduce a digital representation. The word "worm" can be printed arbitrarily many times by using different typefaces and forms of writing but the copies are in a sense (the sense of representing the abstract content "worm") perfect as long as the letters "w", "o", "r", and "m" can be reliably recognised. It is precisely this (functionalist) sense in which digital representations are independent of their physical instantiation, independent of the media. This is how the supposed disembodiment of the digital code starts, and the abstract immaterial existence begins. In general, digital systems are those systems the processes of which can be repeated at will with enough precision and reliability so that the processes can be recognised as being identical with each other with regard to some relevant characteristics.

All definitions of digitality will be contested, and contain different biases, tensions and, consequently, are non-neutral. However, what concerns us here is the ontology of the digital and the effects that ontology may have for the ethical questions around digital information technology. John Haugeland (1998) has offered a useful definition of digitality. A digital device is defined by two processes: first, a read–write cycle and, second, a system of signs in which a type-token distinction can be maintained. A digital device reads and writes tokens of sign types, where the type is distinct from the physical tokens, e.g., this 'a' is a token of the type "a". Like discussed above, the tokens and types must be distinct and mutually exclusive, a sign is either an "a" or a "k", not both. Digitality can be achieved when the read–write cycle of the signs is reliable enough, that is, when the written token of a type is recognised (read) as an instance of the type. Like Haugeland points out, the decisive factor is the reliability of the whole cycle: the writing process can be more lax if reading is precisely controlled and vice versa. So, according to Haugeland, a digital device is composed of three things: (a) a set of types of signs, (b) a set of processes for reading and writing tokens of the types, and (c) a set of conditions under which the read–write cycle is reliable enough.

With technology like this, it is possible to produce perfect copies of sequences of sign types (not of sequences of tokens). In the sphere of the abstract types, a digital copy is perfect, even though all the physical copies (consisting of tokens) in the empirical world are non-identical. Because the level of types can be separated from the level of tokens, the (relative) matter- or media-independence follows. Accordingly, it follows that not everything in the world is digital. A major part of the devices in the world, including a large part of those that have been designed to work digitally, are not digital. Condition (c) is in practice often the one that fails to be satisfied: digital devices get broken because the read–write cycle is not reliable enough. A digital device stops being useful for its intended purpose when through wear and tear it deteriorates to a point where the tokens it presents get blurred so that they can not be reliably recognised anymore.

The distinction between types and tokens is crucial here. If everything matters, even the fact whether an "a" is an 'a' or an 'a', we are dealing with something non-digital. For example, causal relationships in the physical world are non-digital (and analog) in this sense: they are sensitive to the whole continuous scale of initial conditions. The movement of a billiard ball can be *explained* by relying on abstract laws of movement, but the movement *taking place* in the world is sensitive to the initial conditions in a continuous and non-digital way. There is no upper limit to this sensitivity, therefore physical (analog) processes cannot be perfectly copied.

Digital relativity

The issue can be discussed in more detail by considering the question whether everything that is analog

can be digitised. Because the digitalness of a device is always relative to a function or a goal, that is, relative to the reliability needed, the digital is ontologically speaking always dependent on something else and therefore insubstantial or epiphenomenal: we always experience a physical world of particular tokens. The digital is constructed, abstracted or interpreted out of the physical. For instance, when faced with a digital copy of the *Mona Lisa*, we direct our attention only to certain aspects of the light emitted (those that resemble the light reflected from the painting) while disregarding others (including what the light tells of the physical substance emitting/reflecting the light). Some properties of the painting can be digitalised (e.g., properties of the light reflected), while others, like the material doing the reflecting or the unique causal chain from Leonardo's brush to the wall of the Louvre, cannot. The causal chain or the causal properties of the painting cannot be digitalised simply because in those cases we do not know how to separate between the abstract layer of "code" and the physical layer.

Consequently, everything is not amenable to digitalisation. For instance, even if we would agree that some kind of level of code is at work inside some phenomenon, if that code does not support a distinction between types and tokens, digitalisation of the code is impossible. To put the matter bluntly, a hug is not digitalisable, because we do not know what would be the abstract type of which a particular hug is a token. A hug is dependent on unique causal chains, including the physical presence of two persons enduring the same physical risks, threats and promises.[5] This is another way of saying that things that are dependent on particular causal chains and particular times and places can not be digitalised, because they do not support the necessary difference between sign types and sign tokens.

The point is more complex than it seems. The fact that not everything can be interpreted as a system of abstract signs is a contested one. It is possible that everything *can* be described or even explained by using a system of abstract signs. However, it is usually granted that such explanation is based on idealisation and abstraction in the sense that there is an ontological difference between the explanation and the explained: a biological theory of the apple cannot be eaten. In the case of digital code, this ontological difference is blurred by a common view according to which an abstract specification of an algorithm (the explanation) is in some sense more pure or real than any instance of its implementation (the thing explained). However, even in this case the ontological difference prevails in that a digital device does not *work* without the implementation. In the same sense as all linguistic systems border on or include in themselves traces of their non-linguistic preconditions, all digital systems include traces of their physical preconditions, and ultimately succumb to or are erased by something non-digital.[6] Certain devices are able to function digitally under specific favourable conditions. These conditions do not have to degenarate very much for the functioning to become unreliable with regard to the goals.

The existence of the digital is, then, doubly relative. The digital exists with regard to something, in relation to something; namely, with regard to some goals and their reliable achievement as interpreted by us. This is a consequence of the fact that the digital does not exist unless there is an abstract system of signs, which in its turn does not exist unless something is interpreted as a system of signs. First, then, the digital is dependent on a system of signs, which in turn does not exist without interpretation. The tokens have to be seen as tokens of a type. In this way the existence of the digital is relative to the existence of a language and its interpretation, including all the background practices that a language implies. Second, the reliability of the read–write cycle is relative to the goals set. Good examples are provided by the various forms and levels of error–correction and sample rate in digital compact disc technology. Another example could be the way in which counting with ones fingers takes time to become reliable. A child learning the skill has to practice hard in order to consistently distinguish between the fingers and to be able to identify them. To oversimplify, a nine-pixel copy and a nine million-pixel copy of *Mona Lisa* are both digital copies of the painting, but the difference in terms of interpretation and recognition is huge. The necessary condition of reliability depends on the context.

Both of these ontological relativities are significant for ethical questions. The first, the contextuality of the system of signs, is familiar from other fields such as contemporary philosophy of meaning and mind. Nothing in the world as such forces us to regard something as a system of signs. Conversely, practically anything can be read as a language, nothing can stop us from interpreting this or that as a system of signs.[7] The existence of a system of signs and,

[5] This is what Hubert L. Dreyfus implies, but does not spell out, on p. 69 in Dreyfus (2001).

[6] This is why Haugeland's (1998) article is titled "Analog and analog"; in his view the digital is another type or subset of the analog.

[7] John Searle (1980, 1992) is one of the prominent philosophers of mind who has in various ways argued for the view that both the existence of a syntax and of a semantics are relative.

consequently, of something digital, is therefore both under- and overdetermined by the world. Only human interpretation can resolve the matter. The digital, like the syntax and semantics of information processing, are in the eye of the beholder.

The second relativity, the context-dependent reliability of the functioning of the system, characterises the fuzzy border between the analog and the digital. The required reliability and "perfectness" of the functioning are relative to the goal, e.g., that of listening to music. A digital copy of, for instance, a song is never identical or perfect in all respects. It can never in any sensible way reproduce the same vibrations of the same molecules. However, it can be perfect enough so that we are able to recognise and enjoy the song. The "perfect" digital copy is relative in the sense that no string of signs (sign tokens) can be identical to another string of signs (sign tokens). The identity is on the level of idealisation: the strings of signs may make possible the reliable enough recognition of the underlying type, so that an illusion of perfect match is created. Thus the functioning of digital devices is dependent on how well the devices can sustain the human processes of abstraction and recognition.

The existence of these relativities shows that "digital being" includes a whole set of beliefs, a whole world-view, a whole *ethos*. The distinction between tokens and types is powerful and has wide ramifications. As is well known, long-winded philosophical debates can form around the question of if, how and where do abstract entities exist. The classical viewpoints of realism and nominalism readily spring to mind here, too. This philosophical side of the issue has to do with questions of identity and difference. Some differences are abstracted away in order to create perceived identity. An illusion of sameness is created even though the physical world and experience of it are in constant flux. Physical processes are unique in the sense that they cannot be repeated at will. While digital processes can be repeated at will, they are in this sense non-unique.

With a certain rhetorical turn, it could be claimed that "the digital", the belief in the existence of something perfectly repeatable and media-independent that can be owned and operationalised, takes part in producing the truth of the existence of a disembodied and self-identical self (making things like AI, transhumanism, genetic perfectibility *facts* of the present or the future). At the same time, this set of beliefs or forces, this Foucaultian *episteme*, shows its own limits. Unique processes, those connected to particular causal chains and those that do not lend themselves to a separation between types and tokens, cannot, not even in principle, be digitalised. They are not "code", at least not self-inclusive, formalisable, and calculable code. Keeping these relativities of the ontology of the digital in mind, one can turn to the ethical issues. What if the non-digitalisable processes include processes of knowledge that are vital to information societies? What if the appropriation of the digital into the scheme of private property requires (because of the ontological nature of the digital) social formations that are unnecessarily closed or oppressive?

Knowledge creation and ownership of digital code

Digital information technology provides the means for new types of sharing and distributing information as well as the means for unprecedented forms of control and ownership of information. This is the tension included in all the digital utopias and dystopias. Let us take an example of this tension, an example that crystallises the positions. One of the most interesting debates around the ownership of information has been focused on computer software, not the least because software provides extreme examples of both the proprietary and non-proprietary forms of relating to code. On one hand we have the idea and the corresponding practices according to which pieces of computer software are identifiable works that are owned by their authors. Through this we get the proprietary system in which the owners of software, typically software companies, give the user the right to use the software under certain condition specified in the end user license agreements. On the other hand there is the idea according to which computer software is algorithmic also in the sense that it can not be owned: it is created in the interaction of a body of individuals and organisations and is signed in the sense of "the author who discovered something" rather than in the sense "the author who owns something". These extremes are exemplified, for instance, on the level of computer operating systems, where the dominant Microsoft Windows system is the purest example of a proprietary system, and the GNU/Linux system maybe the best-known example of a non-proprietary and free system.

The main economic and social argument for the ownership of software or code in general is that only the economic incentive of property and wealth can guarantee the prosperity of a society. The arguments for the freedom of software are similar to the arguments for the freedom of science: the speed of development, the trustworthiness of the software and the availability of code are all improved by the non-proprietary nature of software, and at the same time

the values of co-operation and sharing are encouraged.

The question can be investigated from the point of view of knowledge intensive work and knowledge production. Both the development of proprietary software in software companies and the development of free software in volunteer "hacker" organisations are certainly instances of knowledge intensive work resulting in goods that are digital information. But the contexts, including the underlying ethical, social and political ideas, are quite different. The underlying sets of beliefs in the different communities are not uniform, to be sure, but they do result in different ethical views.

Software development in a big software company is institutionalised, more or less a Taylorist enterprise with clearly defined and calculable subtasks leading in a controlled manner to a coherent and well-defined result. The knowledge production happens in an organisation with its structure, aims and functions. Such knowledge production has been studied extensively, e.g., in view of the types of implicit and explicit knowledge circulating in the organisation.[8] To use a definition by the sociologist Scott Lash (2002) the functions of an organisation are structured through norms and the power legitimised by the norms. An organisation like this is hierarchical, with the ensuing channels of command and division of labour.

In contrast to this, the development of free software of the GNU/Linux type happens in a widely distributed and non-institutional manner in a global volunteer community co-operating mainly through the Internet. This kind of knowledge creation is not that of the organisational type. Using Lash's terminology, the free software community is a disorganisation based not on norms and rules but on shared values. Lash illustrates the difference between an organisation and a disorganisation with the difference between a church and a sect. The functioning of a church is typically based on a certain hierarchy, a set of norms and legitimised use of power, whereas a sect is formed around a set of shared beliefs or convictions, visions, often exemplified by a charismatic leader. The difference is reflected in the operative force: in an organisation people are controlled and commanded by the use of legitimate power, whereas in a disorganisation people can only be controlled and persuaded through the use of (physical or non-physical) violence. A disorganisation is formed around a set of shared values, therefore the leader (the guru, in the case of Linux, Linus Torvalds) can control the disorganisation by saying what is right and what is wrong, what works and what doesn't, even though the leader does not have any legal power. However, even if a disorganisation is non-institutional and informal, it is not by any means chaotic. It can be much tighter controlled and "organised" than an organisation, because the appropriation of resources does not obey the normativised compartments of an organisation. A sect, a movement, or a tribe can function in a very effective and rational manner. This is one of the elements of free and open source software development that has surprised and continues to surprise economic theory: according to received economic theory there is no real incentive for the volunteers to contribute to free software development.

Against this background the question of ownership of intellectual property (in this case software) can be seen in detail. Software development in a company, in an organisation, can benefit from the fact that pieces of information, knowledge and code are seen as property. The ownership of code forms not only an economic motivation, but also gives a tool for the streamlining, rationalising and economising of the organisational structure, gives a measure for effectiveness and information flows, etc. The propriertarisation and commodification of intangible, abstract code is therefore a real asset to organisational knowledge creation. This also creates an ontological pull: it is good to believe in the independent existence of code. Knowledge creation in organisations gains when code is appropriated in terms of private property, when it is commodified and controlled through systems such as digital rights management.

In contrast, knowledge creation in a disorganisation is at the very least disturbed and at worst destroyed if code is the property of owners. In the GNU/Linux community, the developers take part out of their own will, they "scratch their own itch", and the systematised sharing of information guarantees that their work contributes to a growing, common body of knowledge.[9] The participants can contribute only in so far as the flow of information is not controlled by ownership. The underlying values forming the community include a belief in the freedom of information and a passion for improved information technology. A disorganisation is largely self-organising. The functions of the disorganisation are based on informal and non-institutional co-operation and sharing, that cannot be compartmentalised by decree or rationalised by use of power. The shared values are embedded in the functions of the disorganisation, and those functions are at the same time the goal of the disorganisation.

[8] See, e.g., Nonaka and Takeuchi (1995).

[9] For views on open source and free software development see DiBona et al. (1999) and Raymond (2001).

Disorganisations work as goals in themselves, therefore they do not, in general, benefit from the introduction of external goals. The harm to knowledge creation in disorganisations is done if we believe in the existence of code and assign a proprietary system of ownership for code.

From the point of view of the information society it is crucial to recognise the existence of these two different types of knowledge work, and their different preconditions. A bias is produced by the fact that the visibility of these two processes is different. Organisational knowledge creation is organised also in the sense of having its own effective lobby and representative system, while disorganisational knowledge creation relies on more underhand methods of social and political influence. This bias is all the more unfortunate if there is reason to believe that information societies are increasingly dependent on disorganisations. This is the economico-political side of the issue that, however, cannot be separated from the ethical one. A society where knowledge is asymmetrically divided and fragmented is less equal and democratic than one in which knowledge is considered non-proprietary.[10]

Digital nominalism and the performative contradiction of ownership of code

The example of free software, above, shows that there are voluntary ways of using information technology and "the digital" that deny that the ownership of "intellectual property" would be essential for an information society. These ways of using information technology can be further motivated by making a *reductio ad absurdum* type of though experiment on the idea that information and knowledge should be owned. Let us suppose that all knowledge is property, and that the use of that property is to be controlled by the owner of the property. Let us also suppose that the owners of knowledge want to share it only if there is some gain for them in the sharing. Now, when a child is born into the society she does not know that knowledge is owned. In order to learn that knowledge is owned the child has to be taught a whole lot of things, and she has to live as a part of the society for extended periods of time. The education of the child is impossible if all knowledge is proprietary, unless the owners of knowledge want to "brainwash" the child so that they can gain later on. Legally this would be impossible, however, as a newborn child couldn't enter into agreements that guarantee her compliance after she has learned what "knowledge" is, what "property" is, and the fact that all knowledge is owned. If, however, such agreements are entered only in the mature age, the person can choose to stay ignorant of the fact that all knowledge is owned. She can't learn the fact, if the owners do not divulge the information, or she does not have to learn it, unless she asks from the owners. In all of the cases the presuppositions lead into absurdities: the society in which all knowledge is owned is a closed one.

The main message of the admittedly naïve thought experiment is that knowledge first has to be non-proprietary for proprietary knowledge to be possible. In the same way code first has to be non-digital for digital code to be possible.[11] These are not only temporal priorities but ontological facts. If this is the case, it follows that the utopias of complete digitalisation and proprietary code necessarily demand societies that are closed, because the interpretative system needed to uphold these utopias has to include an ontological commitment to the existence of abstract code. Abstract code is possible based on the affordances that concrete physical things and systems have under favourable conditions. One of the needed favourable conditions is that people are willing to believe in code. Closed social formations are necessary, because an illusion of the reversion of the ontological priority (the idea that the proprietary is more basic than the non-proprietary or the idea that digital code is more basic than physical instantiation) can be created only through extensive social control. Finally, because an interpretative system is always a social one, the belief in abstract, disembodied code may have a negative effect on our understanding of the vital functions of information societies.

Accordingly, we may suggest what types of social practices and structures of power circumvent not only the beliefs in the "uncontrollability" of (digital information) technology but also point to ways of using technology that support equality and openness in information societies. It can be assumed that as an ontological stance digital nominalism, i.e., the belief according to which code is always here and now in particular concrete instances of functioning devices (including biological organisms), and according to which "the digital" is only a *nom-de-guerre* for particular types of electronic devices, directs our ethical and political attention to the issues that are vital for an equalitarian information society. Moreover, it can be assumed that the use of information and

[10] There is also the possibility that a drastically asymmetric division of knowledge produces a less effective and less innovative society: see the argument and the references in Lessig (2001, 2004).

[11] Furthermore, it can be claimed that organisations are based on existing disorganisations: organisations are ways of channeling the creativity of disorganisations; see Lash (2002).

knowledge through the non-proprietary models benefits not only disorganisational knowledge creation, but also a sense of the (digital) technological interests being malleable, not beyond all human control.

References

N. Block. Introduction: What is Functionalism? In: N. Block, editor, *Readings in Philosophy of Psychology*. Harvard University Press, Cambridge, MA, 1980.

J. Derrida. *Negotiations. Interventions and Interviews*. (edited, translated and with an introduction by Elizabeth Rottenberg) Stanford University Press, Stanford, 2002.

C. DiBona, S. Ockman and M. Stone, editors, *Open Sources. Voices from the Open Source Revolution*. O'Reilly, Sebastopol, 1999.

P. Drahos and J. Braithwaite, *Information Feudalism. Who Owns the Knowledge Economy?* Earthscan, London, 2002.

H. Dreyfus, *On the Internet*. Routledge, London, 2001.

J. Haugeland. Analog and Analog. In: Having Thought. Harvard University Press, Harvard, 1998.

S. Lash, *Critique of Information*. SAGE, London, 2002.

L. Lessig, *The Future of Ideas. The Fate of the Commons in a Connected World*. Random House, New York, 2001.

L. Lessig. "Free culture" Presentation at the OSCON 2002 Conference. Available: http://randomfoo.net/oscon/2002/lessig/.

L. Lessig, *Free Culture*. The Penguin Press, New York, 2004.

D.M. Levy, *Scrolling Forward*. Arcade Publishing, New York, 2001.

I. Nonaka, Ikujiro and H. Takeuchi, *The Knowledge Creating Company*. Oxford University Press, Oxford, 1995.

E.S. Raymond, *The Cathedral and The Bazaar*. O'Reilly, Sebastobol, 2001.

J. Searle. Minds, Brains and Programs. *Behavioral and Brain Sciences*, 3: 1980.

J. Searle, *The Rediscovery of the Mind*. The MIT Press, Cambridge, MA, 1992.

R. Stallman. *Free Software, Free Society. Selected Essays of Richard M. Stallman*. Joshua Gay, editor. The GNU Press, Boston, 2002.