Ethics Inf Technol (2009) 11:299–310
DOI 10.1007/s10676-009-9209-7

ORIGINAL PAPER

Convivial software: an end-user perspective on free and open source software

Carl Mitcham

Published online: 3 October 2009

Abstract The free and open source software (Foss) movement deserves to be placed in an historico-ethical perspective that emphasizes the end user. Such an emphasis is able to enhance and support the Foss movement by arguing the ways it is heir to a tradition of professional ethical idealism and potentially related to important issues in the history of science, technology, and society relations. The focus on software from an end-user's perspective also leads to the concept of program conviviality. From a non-technical perspective, however, software is simply a new example of technology, and the effort to assure that technology is developed in a socially responsible manner has a significant history. The argument thus begins with observations about the history of technology. This leads to critical reflections on the development of professional engineering ethics, and to a discussion of the alternative technology movement. Finally, it concludes by indicating some criteria to consider when imagining the design of convivial software.

Keywords Alternative technology · Convivial software · End user · Engineering ethics · Free software · Open source · Technology transfer

The ideals of free and open source software arose within the technical community, although under significant non-technical influences (Turner 2006). While there have been debates among those who would distinguish the free (sometimes *libre*) and the open source movements, since the late 1990s the two regularly fly under the unified acronyms FOSS (free and open source software) or FLOSS (free *libre* open source software), thus reducing the importance of such differences with regard to the present argument. Beyond internalist debates, considerable interest has also development concerning implications for the non-technical or end-user world. Lessig (1999) and Weber (2004), for instance, have analyzed the open source movement from the perspectives of law and political economy, respectively; Chesbrough (2006) has argued the importance for new models of business. Computer scientists, software engineers, and associated hackers have also regularly promoted non-technical benefits across a spectrum of societal dimensions. The free software community has seen the bazaar of open software as productive of both technical and end-user goods in ways that proprietary market software production is not (see, e.g., the collections by DiBona et al. 1999, 2006; Feller et al. 2005). In all such cases, the general approach has been to provide a narrative description of some aspect of the free and open source phenomenon, exploring how what may have initially appeared as non-standard behaviors led to the social construction of new norms with implications for legal, political, or economic orders in ways that are potentially beneficial to society. (Narratives along these lines typically appeal to chronologies that run from the 1950s into the early 2000s, as outlined in the simplified time line provided as an "Appendix".)

Indeed, it is on the basis of such narratives that openness has migrated from software into a plethora of related terms such as "open access" publication, "open content" development (as in Wikipedia), "open innovation", "open course ware", and even "open source culture". Remarkably,

C. Mitcham (✉)
Liberal Arts and International Studies, Colorado School of Mines, 1005 14th Street, Stratton 301, Golden, CO 80401, USA
e-mail: cmitcham@mines.edu

however, there has been little if any linkage between open source ideals and Karl Popper's political philosophy of the open society (Popper 1945), and only quite limited discussion of relations to Robert Merton's classic sociology of how science functions when guided by an appropriate set of scientific norms (Merton 1942), now denominated as "free" or "open science" (see Kelty, in Feller et al. 2005; Kelty 2008). Relations between open source methods and the ontology of open systems (see, e.g., von Bertalanffy 1968) is another connection deserving further exploration.

The present argument will nevertheless take a different tack. The goal here is to begin with the end user and from this perspective to reflect on the free and open source software movement, not so much as it were from the inside out as from the outside in. To do so will involve placing the movement in larger historico-philosophical perspective. Then, going beyond the technical details, the aim will be to enhance and support the free and open source movement, by showing how it is heir to a tradition of professional ethical idealism and potentially related to important issues in the history of science, technology, and society relations. A normative argument will also be advanced for assessing free and open source software in something more than its own conceptions of freedom and openness, but with an appeal to norms that remain inherently complementary.

The approach adopted here has obvious affinities with end-user and user-centered software design analyses (see, e.g., Norman and Draper 1986; Adler and Winograd 1992; Kaasgaard 2000), although it has been argued (see, e.g., Lieberman et al. 2006; Seffah et al. 2005) that such design methods sometimes treat end users as no more than another factor in an on-going technological construction—that is, as customers or consumers—rather than as citizens or true ends for software functionality. Thus, the argument here has more in common with the user-centered perspective of Johnson (1998) or Verbeek (2005) in their analyses of technological artifacts in general; both Johnson and Verbeek, however, adopt rhetorical and postphenomenological stances, respectively, which are kept at a distance. Perhaps the closest affinities are with value sensitive design discourse (see, e.g., Friedman et al. 2006).

The effort to focus on software from an end-user's perspective will also lead to the coining of a new term. From an end user's perspective what is important is not so much the direct availability of source code and the creative freedom to manipulate it as what may be called *program* or *technical conviviality*, phrases deeper and richer in meaning than "user friendly". Related terms that might be proposed include "software without frontiers" or even "humanitarian software"—indicating a desire to break the boundaries of software sovereignty or technicalist nationalism. There is more with regard to openness than program transparency and rights to play in the fields of the code.

From a non-technical perspective, the invention and development of software is no more than another instance of the invention and development of technology broadly construed. Software is simply a new example of technology, and the effort to assure that technology is developed and utilized in a socially responsible manner has a significant history. The argument will thus begin with observations about the history of technology. This will lead to critical reflections on the development of professional engineering ethics, and to a discussion of the alternative technology movement. Finally, it will conclude by indicating some policies or criteria to consider when imagining the design of convivial software.

Technological invention in a social context

The creation of software is a kind of invention. But it is invention with specific problematic features. Writing and adapting source code is an activity more than once removed from end-user utilization, and one that requires high levels of abstract, analytical thinking. It is useful from a historical perspective to consider such features.

Human beings cannot live without technics. By necessity they fabricate clothes and shelter, and they must secure and prepare food. Thus, it is reasonable to re-describe *homo sapiens* as *homo faber*. But just as the thinking of *homo sapiens* is not limited to one kind of rationality—since thinking includes the creative use of language in the arts and the humanities as well as in the analytic and investigative skills of mathematics and science—so too the making of *homo faber* takes many forms. Although necessity may be the mother of invention, she is mother to a large family with many children.

According to philosophers Alfred North Whitehead and Ortega y Gasset writing independently of each other in the early decades of the 20th century, the history of technology went through a critical watershed in the late 19th century, when the process of inventing became rationalized. What Whitehead called "the invention of the method of invention"—that is, the invention not just of another technical artifact but of techniques to promote invention itself—was "the greatest invention of the 19th century", an event that fundamentally transformed technics and the technics-society relationship (Whitehead 1925, p. 141). It replaced craft technics with systematically pursued technology.

Thomas Edison, who has become a kind of inventor archetype—having been awarded a record 1,093 patents—played a key role in this process by inventing what he called an "invention factory" (Israel 1998, p.120). With regard to the motivations driving his invention factory, or research and development (R&D) work, there exists a revealing

story. Edison's first invention was an electric vote recorder for a legislative assembly, developed when he was only 21 and working in telegraph communications—that is, the computer industry of his day. In place of the roll-call or written vote systems, which were slow and time consuming, all members of the legislature would be supplied with electric switches on their desks. This would allow votes to be counted and the results determined quickly.

But this invention was a flop. It did not sell. The Massachusetts legislature, to which Edison offered his electric vote counting device, was not interested. In fact, it was positively opposed. Politicians often wanted to learn how colleagues voted before casting their own votes, or wanted time to reconsider as written votes were manually tallied. In a roll-call vote, members were free to "pass" until others had voted, in order to learn how a vote was going before finalizing their own decisions. Bargaining was possible even in the midst of voting. This possibility would have been curtailed by electric voting. In response to this failure to sell his electric vote-tallying device, Edison determined never again to invent something unless he knew people really wanted it. From now on, he promised himself, he would "devote his inventive faculties only to things for which there was a real, genuine demand" (Dyer and Martin 1929, vol. 1, p. 103). Edison was not enamored of technology practiced solely for what Florman (1976), in an apologia for creative engineering, would subsequently call the "existential pleasures" of technological experience.

What this story reveals is not just an aspect of Edison's business plan, but something about the new systematic invention process itself. The process makes possible what might be called alienated inventing. Karl Marx had previously identified alienation as a key feature of the industrial mass production process. In the large-scale factory structured around extensive division of labor, workers were separated from the full experience of making and from the products of their piecemeal fabrication. Now, in the research and development factory, alienation had moved from technological production into the production of technology. It would give rise to what has become known as the problematics of technology transfer, especially transfer from laboratory to market.

It is difficult to imagine traditional artisans inventing unwanted or truly superfluous products. Traditionally, artisans are so culturally and socially embedded that their inventive skills are just naturally applied to culturally and socially meaningful goods. Their contributions to what might be mistaken for anticipations of a consumer economy were ornamentation and aesthetic decorations that enhance cultural integration by uniting technology, religion, politics, and art. Newness came into existence not suddenly through systematic or conscious pursuit but slowly by give-and-take accretions over extensive periods of time.

In research and development laboratories, however, the great process of separating or dis-aggregating the elements of culture that is a defining characteristic of modernity (Polanyi 1944), took hold of the invention process and disembedded it from traditionally integrated socio-cultural orders. As one critical social theorist has put it, prior to the modern period

> Technology was associated with a way of life, with specific forms of personal development [and] virtues…. Only the success of capitalist de-skilling finally reduced these human dimensions of technique to marginal phenomena. (Feenberg 1995b, p. 18)

The Renaissance, Reformation, and Enlightenment introduced distances among, for example, art, religion, politics, and economics—all of which became semi-autonomous social activities, whose integration henceforth had to be constructed, not assumed. So too the invention process was now uprooted from all implicit connections with a socio-cultural nexus and on a macro level de-contextualized, which is not to deny that at individual or micro levels technical activities remain enmeshed with and influenced by individual motives and interests. Because Edison had no tacit or intuitive knowledge of political life, he invented something that did not fit in with that life—and vowed in the future to do what came to be called market research prior to any inventing.

For a time, the new invention of invention method could rely on the residual memory of contextualized human need. Edison's major inventions did not required any systematic market research to be successful. Electric lights, voice recording, and motion pictures all responded to almost mythical human desires—although many of these primordial desires and possible device-meeting responses had also been mythologically criticized. His great techno-economic innovation of the electric power system was a direct response to popular enthusiasm for electric lights. But henceforth, especially in particulars, cultural and social contexts of use would increasingly have to be investigated in advance by means of empirical research—or created by means of advertizing. In the process such research and advertizing had the effect of forming a new socio-economic order in which inventing could take on a kind of self-sustaining character.

Shortly after Whitehead, Ortega y Gasset (1939) rightly noted, without reference to Edison, that once there is in place a dis-embedded method of invention, although inventors may initially tend to become alienated from the traditional lifeworld, they subsequently begin to transform human affairs through their infusions of inventiveness. Inventors do have a tendency to form a culture of interest in

invention for its own sake, to think of themselves as nothing more than inventors, sucked into the vortex of the existential pleasures of invention—at once depriving themselves of the common pleasures of social existence. They turn "art for art's sake" into "technology for technology's sake", at the same time inventing a new macro socio-cultural order. Divisions between expert and citizen, producers and consumers—dimensions deeper than mere electronic digital divide product accessibility—open up at the feet of technoscientific specialists and behind the backs of democratic end users. Such divides can only be bridged or re-united through the systematic development of ways to relate the different worlds.

Comparisons with software development are no doubt obvious. The hacker culture has a tendency to become turned in on itself. Hackers are defined by their technical skills and their delight in problem solving, as well as by their commitments to share solved problems. Reinventing the wheel is not only a waste of time and energy, it is not fun. What is fun is inventing something new, and then sharing the excitement. Software engineers write code that is technically sweet and pleasurable.

Having posted Linux kernel version 0.01 without fanfare a month earlier, here are the words of Linus Torvalds, another 21-year-old inventor, when he announced version 0.02 (October 5, 1991):

> Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an [operating system] you can try to modify…? No more all nighters to get a nifty program working? Then this post might just be for you.

In the midst of his technical delight there is surely a measure of separation from end-user interests. In an earlier email (August 25, 1991) Torvalds wrote, "Any suggestions are welcome, but I won't promise I'll implement them". In a later book, he elaborated on "the meaning of life" by distinguishing three basic human motivations: survival, social relations, and entertainment. "Taking survival for granted", Torvalds argued that Linux "brought people both the entertainment of an intellectual challenge and the social motivations associated with being part of creating it all" (Torvalds and Diamond 2001, p. 249)—the existential pleasures of creating software "just for fun".

For Torvalds, of course, the implicit invitation is also for other technically sophisticated users to take pleasure in doing some implementation themselves. But this is a technical energy and excitement that commercial powers readily seek to harness and direct. As even the greedy Microsoft has on occasion reminded, the proprietary commercialization of software did not originate solely out of greed. Yet commercialization has its own downside. The market must itself be created and designed to benefit the common good, because it is in constant danger of being captured by private or special interests. The free and open source movement is one attempt to shift the balance of power in the market economy, but it is a shift that may call for supplemental adjustments among those who participate in this type of engineering creativity and its entrepreneurial follow-ons.

The engineering ideal

During the same period that Edison was inventing the method of invention, engineers were inventing their profession—an engineering ethos. Edison was not himself a professionally trained engineer; like many hackers today, he was an autodidact. So were many engineers of the time. And just as Edison created a method of invention, so other engineers were in the process of creating a method for producing engineers, one that would systematize the process of technical self-education.

The invention of professional engineering and its correlate, a professional engineering ethos, later articulated in part as an engineering ethics, exhibits its own complex history. Here it is sufficient to note how the invention of engineering education was a response to the mass need for engineers brought about by mass production and mass invention. As part of his creation of the invention factory, Edison himself began to employ increasing numbers of engineers and to support engineering education. Moreover, the emergence of professional engineering ethics may be read as a parallel response to Edison's technology transfer problem, the difficulty of coordinating technological invention and design with end-user interests. Instead of the commercial responses of market research and advertizing—or Edison's own personal commitment—the engineering professional began to construct institutional commitments to address such dangers.

The emergence of engineering ethics may be conveniently sketched in terms of a three-phase argument. (The following narrative draws on one previously explored in, e.g., Mitcham 1994.) The first phase or argument occurred during the 1700s and the 1800s as civil engineering emerged from the military. The first engineers were military engineers, designers of fortifications and designer-operators of "engines of war" such as catapults and battering rams. The first schools to award engineering degrees were established by the military or some agency of the state. As engineers emerged out from under the shadows of the military to become civil engineers, they established professional engineering societies. In England, as an outgrowth of informal dining clubs, there emerged the Society of Civil Engineers in 1771, then the Institution of Civil Engineers in 1818, which

was granted a royal charter in 1828. The Institution of Mechanical Engineers followed two decades later.

When granted its Royal Charter the Institution of Civil Engineers asked one of its members, Thomas Tredgold, to formulate a definition of engineering. The result is the now classic description of engineering as "the art of directing the great sources of power in nature for the use and convenience of [humans]". But the truth is that engineers do not work for human beings in general; most work for private corporations, and thus easily become captives of the organizations that pay their salaries. This is different from such professionals as physicians and lawyers, who mostly run their own companies—or often work directly for the state. The ethical results are obvious: Engineers lack the level of professional autonomy enjoyed by physicians and lawyers. Indeed, having arisen out of the military, there is a tendency for engineers to adopt some form of military ethics. The ethics of soldiers is obedience to authority. The implicit ethics for civil engineers tended toward company loyalty. That is, in the first instance, engineers quietly assumed, despite the ultimate ideal of serving humanity, that they owed obedience to their employers.

A second phase or argument concerning engineering ethics occurred in the early 1900s. Professional engineering societies drafted the first explicit professional engineering ethics codes. These codes simply made explicit what was already implicit: the importance of company and professional loyalty. But the inadequacy of this position was immediately felt. Physicians had a moral obligation to promote health. Lawyers at some fundamental level were guided by the moral ideal of justice. Neither physicians nor lawyers saw themselves as able to be told what to do by patients or clients; instead, they sought to mediate to patients or clients a substantial ideal that would otherwise be denied them and to which the professionals saw themselves in ultimate serve. Was there no substantive ideal to which engineers, as well, were in service?

During the early 1900s in the United States proposals were debated concerning the possibility that for engineering the substantive ideal was efficiency. The problem with this proposal was two-fold. First, it failed to acknowledge the extent to which efficiency is context dependent, that is, varies with what outputs and inputs are to be considered in calculating any output-input ratio. Second, it tended to justify some form of technocratic governance, thus undermining democracy.

Finally, in a third phase or argument, the inadequate ideals of loyalty and efficiency were replaced by a new formulation of engineering responsibility. Immediately following World War II, professional engineering ethics codes began to argue for "public safety, health, and welfare" as the primary professional engineering ideal. Over the course of five decades, from the late 1940s to the late 1990s, this argument became the consensus view. A philosophical restatement of the trajectory and its projection is that engineers have a duty *plus respicere*, to take more than engineering into account (Mitcham 1994, p. 164).

This consensus is reflected in the "Software Engineering Code of Ethics and Professional Practice" (1999), which declares that software engineers shall adhere to eight principles "in accordance with [a general] commitment to the health, safety and welfare of the public". (Software engineers just make a minor shift by exchanging the order of safety and health.) Indeed, the first of these principles is that "Software engineers shall act consistently with the public interest".

By the beginning of the 21st century, knowledge of such codes and reflection on the issues associated with their implementation and practice, had become a required part of the engineering education curriculum. According to the Accreditation Board for Engineering and Technology, subsequently ABET—the independent professional accrediting agency for all engineering education programs—there are eleven required outcomes for any branch of engineering education. One of these outcomes, along with knowledge of mathematics and science, and skill in design, is "an understanding of professional and ethical responsibility" (ABET 1998).

The free and open source software movement constitutes a major opportunity for the exercise of this engineering ideal of responsibility. Additionally, since one of the more common ways to teach engineering ethics is through case studies, the free and open source software movement may be used as a good illustration of efforts by engineers to take ethics into account. It can easily be argued that open source is a necessary means for protecting public safety, health, and welfare in general, and for practicing at least two of the eight principle specifications. Principle three of the "Software Engineering Code of Ethics", for example, states that "Software engineers shall ensure that their products and related modifications meet the highest professional standards possible". As Eric Raymond has argued at length in *The Cathedral and the Bazaar* (1999), in many instances this requires open source code availability. Principle five further declares that "Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance". It is difficult to see how this is possible without open source code and the freedom to alter and adapt such code. Thus, a strong argument can be made that the "Software Engineering Code of Ethics" entails support for the free and open source software movement. Indeed, it could be argued that the even more general "Association for Computing Machinery (ACM) Code of Ethics and Professional Conduct" (1992) points in the same direction.

But to make public safety, health, and welfare a key value also suggests the importance of the public—and raises again the question of the role of the end user. In the various fields of applied ethics—of which computer and software ethics are only two instances—there have been subtle if measurable movements toward involving the public in technical decision making. (See, e.g., discussions of collaborative design, as examined in Scrivener et al. 2000.) There are, of course, strong resistances to this idea from many quarters. After all, the argument goes, how can non-experts tell experts what to do? Will public interference not undermine professional technical autonomy? Is there not a danger of politicizing science and engineering?

Despite the dangers, there are nevertheless three overlapping arguments for promoting the proper involvement of the public in technical decision making (Mitcham 1999). A first is that public involvement promotes democracy and public intelligence. Insofar as citizens let experts decide for them, they cease to function as active members of the body politic; but insofar as they work with experts to help make decisions about the proper goals for scientific research and technological development, they learn about science and engineering, their powers and their limitations, and enhance their lives as democratic citizens.

A second argument is that public participation may also promote better technical decision making. All knowledge is not technical knowledge. Local knowledge, non-expert knowledge, what is often called indigenous knowledge, when appropriately utilized, has the power to enhance even the technical aspects of technical decisions. One good example comes from HIV/AIDS research (see Feenberg 1995a, Chap. 5). It was HIV/AIDS activists who helped increase funds for such research and redirected the research away from basic work on immunology and toward more immediately practical therapies for those afflicted with the disease. Without the involvement of non-technical HIV/AIDS activists, not as many lives would have been saved by the related scientific research.

Finally, a third argument is simply that people have a prima facia right to influence those decisions that affect them. Although there are clearly instances in which this right may be trumped by other considerations (see Nozick 1974, p. 268 ff.), the public implementation of technological design decisions are arguably not among them. Many scientific and engineering decisions have major impacts on our ways of life. Democratic revolutionaries once legitimately proclaimed, "No taxation without representation". Today, as Goldman (1992) has argued, they may just as legitimately argue, "No invention without representation". This extends the principle of free and informed consent as generally accepted in medicine to engineering (see Martin and Schinzinger 1983).

Indeed, one of the clearest instances of a shift from "technical professional knows best" to "technical professional consults best" has occurred in the medical profession. Traditional medical ethics emphasized the physician as decision maker for the patient. Today this model is in transition to one in which the physician helps educate patients so that they are able to make free and informed decisions about the course of treatments they may or many not undergo. Physicians may be experts but are not by that reason alone the final authority. The practice of medical expertise requires collaboration with end-user patients who themselves determine how that expertise is finally to affect their lives.

The convivial technology ideal

The Edison vote recording machine anecdote illustrates the problem of technology transfer from laboratory to market. In this context, the separation between invention and end-user need is intensified and rationalized by notions of technical autonomy and the existential pleasures of cutting-edge invention, with the responses of marketing and advertizing not so much revealing the problem as obscuring it. The development of personal and professional engineering ethics, insofar as professional ethics calls for commitment to a public good, is another way to address this obscure problem.

But the problem of alienation from end-user concerns is refocused by another type of technology transfer, that from one country or society to another. Since the middle of the 20th century this second type of technology transfer has played an increasingly important role in world history.

On January 20, 1949, in his inaugural address, Harry S. Truman, the first post-World War II President of the United States, declared that having used science and technology to win the greatest military conflict in history, the U.S. now had a new responsibility: "We must embark", Truman declared, "on a bold new program for making the benefits of our scientific advances and industrial progress available for the improvement and growth of underdeveloped areas" (quoted in Sachs 1992, p. 6). With these words and subsequent actions, Truman not only inaugurated a presidential term, but the "age of development". Ever since the U.S. and its allies have systematically attempted to transfer technology from developed to underdeveloped countries, in order to bring all societies into a common framework. Twenty-first century discussions of economic globalization are but the most recent permutations of development ideology.

Yet by the mid-1970s it had become obvious that the initial development strategies for transferring advanced technologies from one country to another often failed for

reasons similar to those experienced by Edison with his vote tallying device. What donor countries wanted to give, recipient countries often resisted taking or tried ineffectively to take. Water and electrical power systems broke down; new roads and machines were not maintained. Gaps between the rich and the poor were not reduced, but widened.

Four responses emerged: One response was to promote social science research into the problem in order to enhance the development bureaucracy. The resulting products often functioned as international marketing studies.

A second response was to promote education as a required adjunct to international technology transfer. The education in question, however, often functioned more like advertising than anything else. In many cases it simply facilitated immigration of a newly trained technical elite from poor to rich countries, a kind of social capital exploitation in parallel to natural resource exploitation.

A third response was simply to cut back on development aid, in the belief that recipient countries simply had to do it for themselves. As one book proclaimed "Underdevelopment Is a State of Mind" (Harrison 1985)—referring, of course, to the state of mind of the poor, not the rich.

The fourth response was proposals for what were variously denominated as intermediate, appropriate, or alternative technologies. Intermediate technologies were designed to function in between manual tools and complex high-tech machines. One example: replace animal transportation not with the internal combustion engine, but the bicycle. Such intermediate technologies were also described as more appropriate to the cultural and social contexts because they introduced smaller, less destabilizing technical changes, and their functioning was easier to understand. The workings of a bicycle are much more transparent than those of an internal combustion engine.

In a reflection back into the high-tech world itself, especially after the energy crises of the 1970s and in the face of mounting issues of environmental pollution, such technologies were seen as appropriate not just to underdeveloped or developing countries but also to highly developed countries. Now they were called alternative technology. The soft energy paths of renewable energies—wind and sun instead of coal, oil, and nuclear power—were argued to be better for both the poor and the rich, and a new basis for creating common frameworks of development (Lovins 1977).

The most influential statement of the intermediate and appropriate technology movement was E.F. Schumacher's *Small Is Beautiful* (1973). Although Schumacher's book is the most widely known, another provides a deeper analysis: Ivan Illich's *Tools for Conviviality* (1973). Indeed, for a short period of time the term "convivial tools" or "convivial technology" supplemented those of appropriate, intermediate, and alternative technology. Although the term has been sidelined by history, it is revived here to suggest again broader ways of thinking about the free and open source software movement.

In *Tools for Conviviality* Illich opened by noting the following about many technologies: They begin as means to some specific end, but often wind up subverting that end through what he calls "counterproductivity". Illich's case study example was medicine. What started out as a means to health has become the cause of a new kind of illness: iatrogenic or physician-caused illness and disease. According to Illich, a first watershed in the history of scientific medicine occurred when it came to pass that physicians actually did something beneficial for their patients more than 50% of the time. This watershed occurred during the early decades of the 20th century. But a second watershed occurred when in came to pass that more than 50% of the time physicians treated patients for illnesses or diseases to which medical practice itself contributed. Such illnesses and diseases range from the results of overt malpractice to the negative side effects of drugs, infections caused by treatments, diseases caught in hospitals, diseases that did not even exist until the evolution of bacteria in response to antibiotics, and more. According to Illich, this second watershed had been reached or was on the verge of being reached at the time his book was written.

For Illich, the proper response to the counterproductivity of the second watershed is moderation: not more medicine, but less. Convivial technology is properly limited technology, technology that does not so overwhelm with its presence that people think they cannot live without it. This is an argument that has been reiterated in the course of more than one critical examination of the contemporary healthcare system (see, for instance, Callahan 1998).

As a response to the challenge of transferring technology from one country to another, then, Illich's argument called for the exercise of critical distance. Perhaps not all technology should be transferred. Prior to attempting a transfer, rather than doing a marketing study of what people wanted, it would be better to do an assessment of the technology itself. Not all technologies are inherently beneficial. Many technologies bring with them political implications if not unintended negative consequences that deserve careful consideration (see Winner 1980). What is more important than technology itself is what we might term, in a modestly ironic adaptation of Illich, its conviality quotient. To what extent does a technology make possible or perfect human living (*vivere*) with (*con*) its artifactual presence?

The appropriation and adaptation of the term "conviviality" in this context might well be questioned as failing to appreciate fully a Kantian-like personalism at the core of Illich's normative stance. At one point, for instance, Illich writes that he means "convivaility" to indicate

"autonomous and creative intercourse among persons, and the intercourse of persons with their environment; and this in contrast with the conditioned response of persons to the demands made upon them by others, and by a man-made environment". He considers "conviviality to be the individual freedom realized in personal interdependence and, as such, an intrinsic ethical value". Indeed, he affirms that "in any society, as conviviality is reduced below a certain level, no amount of industrial productivity can effectively satisfy the needs it creates" (Illich 1973, p. 11). One way of interpreting this claim is to see Illich as affirming individual autonomy of the will over and against any heteronomy, especially in the use of tools or technologies. Human beings are diminished in individual human flourishing insofar as their use of technologies is determined by engineer designers, capitalist producers, advertizing marketers, or any other external managers. Like de Certeau (1980) and Feenberg, Illich would seem to be arguing on the basis of and celebrating creative consumer transformation of technologies.

Although there is clearly some appeal to individual autonomy in Illich's normative argument, it is a mistake to emphasize this at the expense of its communitarian element. According to Illich's anthropology, humans are human insofar as they live with others. In the quoted passage, for instance, Illich references autonomous creativity in relation to "intercourse among persons, and the intercourse of persons with their environment". In his first justification of the term, Illich writes that the convivial society is one "in which modern technologies serve politically interrelated individuals rather than managers" and that "convivial" is meant as "a technical term [designating] responsibly limited tools" (Illich 1973, p. xxiv). The emphasis is on "*interrelated* individuals", not simply individuals. In the only other text with an extended discussion of the term, Illich likewise emphasizes, along with autonomy, that "convivial tools" as those "which facilitate the individual's enjoyment of use-values—without or with only minimal supervision by policemen, physicians, or inspectors" (Illich 1978, p. 42). But again, Illich's anthropology is one that understands individuals as living with others, convivially, among friends. His position is less sympathetic to libertarian individualist rights and more to groups rights as argued, for example, by Michael Ignatieff in *The Rights Revolution* (2007). A convivial assessment of technologies could thus be postulated as one that considers critically, to venture a provocative analogy, to what extent a tool-human grouping might tend to be broken apart and users separated from promised or projected use-values simply as a result of artifactual design.

As one example of such an assessment, consider how the shift from tools to machines to computerized devices tends inherently to diminish human-artifact integration. Tools require human energy input and guidance. A hammer is held in the hand, has energy imparted to it from the muscles of the arm, and is guided by the informed eye, which in turn reforms the motion of the hand. The hammer functions as an extension of the human body.

A machine, by contrast, receives energy and movement from non-human sources, although the human remains as an immediate guiding operator. The automobile is powered by an internal combustion engine but requires a human driver. The form of conviviality or "living with" manifested in the car is quite different than that experienced presented by the hammer. The car begins to take on a kind of independence or semi-autonomy—if the driver unhands the steering wheel, the automobile can continue along the road on its own. The car is less an extension of the human body and more an energized artifact standing over against the body, if not a capsule that temporarily encloses and transports the body.

In computerized machines this trajectory toward living more *away from* than *with* reaches a new level. Computerized devices such as automated teller machines are powered not by human energy but by electricity and guided not directly by human beings but by stored programs. Human guidance exists behind the scene, of course, but so deeply hidden or removed that the artifact appears to take on an independent life. To live with automatic teller machines becomes less and less a direct bodily engagement, as with tools, and more and more a living in the midst of, as with plants and animals. Technical conviviality exhibits distinctive experiential structures.

A further investigation of this issue of technical conviviality can be found in the work of Albert Borgmann, another philosopher deeply concerned with the interactions of technology and culture. According to Borgmann (1984), the separation between inventor and end user is reified in the material artifact, in the form of a separation between the machinery of a device and the commodity it delivers. The commodification of culture is not just the creation of an economy of quick drop shopping, but one in which the goods shopped are commodities whose operations are fundamentally opaque.

Without the details, one can understand how a mechanical, analogue clock works. Without the details, one has hardly any clue about what is going on in an electronic, digital watch. Both deliver a commodity we might call well-tempered time. But in the former, the mechanism of delivery is imaginable even when not fully understood. In the latter, the mechanism disappears behind a veil of technical sophistication. The digital divide becomes an ontological feature of high-tech artifacts.

Something similar happens in the shift from typewriter to word processing computer. Even if one cannot completely master the mechanics of a mechanical typewriter,

its structure and how it works—the transfer of motion from keyboard to precast typewriter key—is readily comprehensible. In living with the typewriter one's intelligence extends itself with ease into the world of artifice. One does not feel dumb or stupid, or come up against an opaque construction that belittles one's common cognitive abilities. Instead, looking at and using the typewriter gives one a certain satisfaction. One quickly develops a sense of competence and confidence in its presence.

In the presence of word processing computers, by contrast, almost the opposite seems to be the norm: one feels stupid, even angry, unable to understand what is really going on, and often unable to cope. Would it be possible for the free and open software movement to address such a problem? Could the free and open source movement design a word processing program that, like the mechanical typewriter, promoted rather than inhibited technical conviviality? And would such software not be inherently more available for transfer from one country or context to another, in the way that bicycle technology is more transferable than automotive engineering?

Conclusion and implications

The history of technology and Edison's invention of the method of invention have been used to point up a problem in technology transfer from laboratory to market. The problem arises from a temptation—a temptation from which the software community is not immune—to become an elite removed from the non-technical end-user life that it is meant to serve.

The argument then reviewed the development of engineering ethics as a partial response to the problem. The contention was that the professional ethical commitment to public safety, health, and welfare supports the free and open source software movement.

Finally, reflection turned to the challenge of development and the issue of technology transfer from rich nations to poor, in order to highlight ways in which the free and open source movement might be thought of as another form of the intermediate, appropriate, and alternative technology movements.

By way of conclusion, it is appropriate to offer a brief statement of possible policy implications. As noted at the outset, from the end-user perspective what is important is not so much the direct availability of the source code as program or technical conviviality, phrases richer in significance than "user friendly". Such is the background for proposing the concept of convivial software. From the perspective of end users, there is more with regard to free expression or openness than technical source code transparency.

To provide concrete illustration this *more*, perhaps it can be suggested that convivial software be judged in terms of three criteria:

1. *Stability*: The need for continuous upgrades detracts from technical conviviality. It is like having friends who insist on continuously reinventing themselves. It is difficult to get comfortable with ever-new friends and software. A missing good in software is the stability of the old—forcing one to repeatedly upgrade. As a personal example, since I first started using Word Perfect in the mid-1980s I have been forced into the upgrade and re-learning cycle at least five times—on average, once every 3 or 4 years. Compare this with my typewriter. I bought a typewriter when I was in high school and used it continuously, with only minor repairs and cleaning, for 30 years. The shift from typewriter to word processor was a positive improvement in writing tools. No upgrades in software have ever come close and, in fact, many have been negative.
2. *Transparency*: One cannot readily figure out how a word processor works. This is true at multiple levels. At one level, opacity occurs in the toolbar icons, which the context sensitive help notes seldom resolve, and in the customizing of default configurations. At another, it concerns what happens behind the scenes, as it were, with formatting. Word Perfect at least allows users to reveal formatting codes, and then edit them. Microsoft Word takes this ability away from its users, and is thus much more mysterious and difficult to adjust. At still a third level of transparency, users might like to understand what happens when a word processing program locks up, requiring the computer to be turned off and re-booted. The error messages are completely opaque. (Indeed, it is difficult to image that the error messages mean very much to the technical expert without some reference materials at hand.) Would it be possible to design a word processing software program that could be understood without earning a degree in computer science and engineering?
3. *Simplifiability*: Most word processing programs include many more features than any one user actually needs. The toolbar is cluttered. I never use "Quick Format" or "Picture Draw" capabilities, and have met few people other than technical users who do. When users need such features they could install specific programs. I would also willingly dispense with "Zoom" and "Change View". But it is not possible to get rid of these features or even to hide them as a way to simplify the screen.

In response to the obvious criticism that one does not have to use all the features built into a word processing program: Yes, this is true. But they clutter up not just the

screen but one's technical life with artifacts. Some of us would like to live with a computer screen that has the aesthetic feel of Shaker or Scandinavian furniture. Why is this not a digital option? Is it not one that the free and open source software movement for the first time might make possible for all?

In summary, then, the argument here is that the free and open source software movement might well adopt and promote an ideal of convivial software that is stable, transparent, and simplifiability. To restate in other words, consider the ideal freedoms zero through three as described by the Free Software Foundation in 1986:

Freedom 0	The freedom to run the program for any purpose
Freedom 1	The freedom to study how the program works, and change it to make it do what you wish
Freedom 2	The freedom to redistribute copies so you can help your neighbor
Freedom 3	The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits

Might it not be possible to complement such freedoms with three others, two negative and one positive:

Freedom 4	Freedom from upgrades
Freedom 5	Freedom from technical study as a prerequisite to basic understanding
Freedom 6	Freedom to simplify (not just improve)

Upgrade free, transparent, and simplified software would be able to help mediate the technology transfer divides between laboratory and market, and between developed and developing countries—and maybe even transform our own development. Free and open source software has the prospect of being able to contribute not just to technical power but to the good life.

Finally, the pursuit of the concept of conviviality in open source software might well have implications for other forms of open source culture—from open access to open science. That is, rather than focusing on the internal structure of knowledge production and then working out to the social implications, it might be reasonable on occasion to reverse the analysis, to work from what would be most useful to humans back toward embedding end user concerns in the knowledge production processes themselves. The normative concept of conviviality, bringing a notion of how best to live to bear on the means of living, may have benefits that go beyond computers and information technology.

Acknowledgments This article grows out of a presentation delivered initially at the "Conferencia Internacional de Software Libre: Open Source International Conference" in Málaga, Spain, February 18–20, 2004. An earlier, less developed version appeared as "El software convivencial", *Argumentos de Razón Técnica*, issue no. 10 (2007), special issue on "Eticas y políticas electrónicas: Ciudadanía digital, software y conocimiento libres", pp. 19–41. Thanks to Andoni Alonso and two anonymous reviewers for comments and criticisms that improved the arguments here.

Appendix

The free and open source software movement: a selective time line

(Freely adapted from www.opensource.org, www.openknowledge.org, wikipedia, etc.)

1950s Software is normally distributed with source code and without restrictions.

1960s Hacker culture, emphasizing openness and sharing of software development, begins to emerge in academic computer science laboratories at Stanford, Berkeley, Carnegie Mellon, and MIT.

1969 ARPANET software is made freely available, which is a key factor in its transformation into the Internet.

1970s Source code remains freely distributed among academic computer scientists and engineers.

1978 Donald E. Knuth (Stanford Univ.) begins work on TeX, a typesetting system, which is distributed as free software.

1980 Scribe (CMU text-formatting program) is privatized; so is LISP (MIT).

1983 Richard Stallman writes the GNU (Gnu's Not Unix) Manifesto, calling for a return to the public sharing of software and source code.

1984 Stallman resigns from MIT to create a GNU as a Unix-compatible free software operating system.

1985 Stallman sets up the Free Software Foundation (FSF) to support his and others' work.

1989 Stallman creates the GNU General Public License (GPL) that defines copyleft protections. Copyleft allows anyone to freely use, modify, and distribute the software, but restricts their ability to copyright it or any modifications they make.

1990 Stallman is awarded a MacArthur Fellowship and begins work on a GNU kernel, HURD.

1991 William and Lynne Jolitz (Univ. of California Berkeley) describe how to port BSD (Berkeley Software Distribution) Unix to i386-based PCs, making possible a free BSD operating system.

Linus Torvalds, a student at the University of Helsinki, releases a Unix-like open source kernal called Linux (Linus + Unix). See Torvalds and Diamond, *Just for Fun: The Story of an Accidental Revolutionary* (New York: Harper Collins, 2001).

1992 U.S. Air Force awards New York University a contract to build an open source compiler for Ada 95, which is based on GNU and called GNAT.

1994 Marc Ewing begins the Red Hat GNU/Linux distribution.

First issue of *Linux Journal* appears.

1996 First conference on Freely Distributable Software, Cambridge, Massachusetts, USA.

1997 Raymond publishes "The Cathedral and the Bazaar", arguing that open source licenses result in higher quality, less expensive software. (The essay subsequently becomes the title essay of a book, 1999.)

1998 Netscape announces that it will open the source code for Netscape Navigator 5.0.

Eric Raymond, Bruce Perens, and Tim O'Reilly, arguing that the free software movement needs better articulation, form the Open Source Initiative to certify free/open source licenses.

Linus Torvalds and Linux appear on the cover of *Forbes* magazine (Aug. 10).

1999 Red Hat Linux and VA Linux go public.

2002 Junta de Extremadura (Spain) creates Linex as the operating system for its computer network, and distributes Linex for public use. Among other governments that have since adopted free and open source software as basic operating systems are Singapore (2004), Venezuela (2004), Peru (2005), Dominican Republic (2008), and Ecuador (2008).

2004 Publication of Steve Weber's *The Success of Open Source* (Cambridge, MA: Harvard University Press).

2005 Software Freedom Law Center (SFLC) founded to provide legal counsel to advance the Free and Open Source Software movement.

References

ABET. (1998). *Engineering criteria 2000.* The most recent version of this document, re-dated for the current academic year, is available as "Criteria for accrediting engineering programs", at www.abet.org. The original document is included as an appendix in Lattuca L. R., Terenzini P. T., & Volkwein J. F. (2006). *Engineering change: A study of the impact of EC2000*. Baltimore: ABET.

ACM. (1992). Association for computing machinery code of ethics and professional conduct. Available at www.acm.org.

ACM. (1999). Software code of engineering and professional practice. Available at www.acm.org.

Adler, P. S., & Winograd, T. A. (Eds.). (1992). *Usability: Turning technologies into tools*. New York: Oxford University Press.

Borgmann, A. (1984). *Technology and the character of contemporary life: A philosophical inquiry*. Chicago: University of Chicago Press.

Callahan, D. (1998). *False hopes: Overcoming the obstacles to a sustainable, affordable medicine*. New York: Simon and Schuster.

Chesbrough, H. (2006). *Open business models: How to thrive in the new innovation landscape*. Cambridge, MA: Harvard Business School Press.

De Certeau, M. (1980). *L'Invention du quotidien: Arts de faire (Vol. 1)*. Paris: Union Générale d'Editions. English version: *The practice of everyday life*, trans. Steven Rendall (Berkeley, CA: University of California Press, 1984).

DiBona, C., Ockman, S., & Stone, M. (Eds.). (1999). *Open sources: Voices from the open source revolution*. Beijing: O'Reilly.

DiBona, C., Cooper, D., & Stone, M. (Eds.). (2006). *Open sources 2.0: The continuing evolution*. Beijing: O'Reilly.

Dyer, F. L., & Martin, T. C. (1929). *Edison: His life and inventions* (2nd ed., Vol. 2). New York: Harper.

Feenberg, A. (1995a). *Alternative modernity: The technical turn in philosophy and social theory*. Berkeley: University of California Press.

Feenberg, A. (1995b). Subversive rationalization: Technology, power, and democracy. In F. Andrew & H. Alastair (Eds.), *Technology and the politics of knowledge* (pp. 3–22). Bloomington, IN: Indiana University Press.

Feller, J., Fitzgerlad, B., Hissam, S. A., & Lakhani, K. R. (Eds.). (2005). *Perspectives on free and open source software*. Cambridge, MA: MIT Press.

Florman, S. (1976). *The existential pleasures of engineering*. New York: St. Martin's Press.

Friedman, B., Kahn, P. H., Jr., & Borning, A. (2006). Value sensitive design and information systems. In P. Zhang & D. Galletta (Eds.), *Human–computer interaction in management information systems: Foundations* (pp. 348–372). Armonk, NY: M.E. Sharpe.

Goldman, S. (1992). No innovation without representation: Technological action in a democratic society. In S. H. Cutcliffe, S. L. Goldman, M. Medina, & J. Sanmartin (Eds.), *New worlds, new technologies, new issues* (pp. 148–160). Bethlehem, PA: Lehigh University Press. Spanish version: Ninguna innovación sin representación: La actividad tecnológica en una sociedad democrática. In J. Sanmartin, S. H. Cutcliffe, S. L. Goldman & M. Medina (Eds.), *Nuevas Mundos, Nuevas Technologies, Nuevas Perspetivas* (pp. 269–286). Barcelona: Anthropos.

Harrison, L. E. (1985). *Underdevelopment is a state of mind: The Latin American case*. Lanham, MD: University Press of America.

Ignatieff, M. (2007). *The rights revolution* (2nd ed.). Toronto: Anansi Press.

Illich, I. (1973). *Tools for conviviality*. New York: Harper and Row.

Illich, I. (1978). *Toward a history of needs*. New York: Pantheon.

Israel, P. (1998) *Edison: A life of inventions*. New York: John Wiley

Johnson, R. R. (1998). *User-centered technology: A rhetorical theory for computers and other mundane artifacts*. Albany, NY: State University of New York Press.

Kaasgaard, K. (2000). *Software design and usability: Talks with Bonnie Nardi, Jakob Nielsen, David Smith, Austin Henderson and Jed Harris, Terry Winograd, and Stephanie Rosenbaum*. Copenhaged: Copenhagen Business School Press.

Kelty, C. M. (2008). *Two bits: The cultural significance of free software*. Durham, NC: Duke University Press.

Lessig, L. (1999). *Code: And other laws of cyberspace*. New York: Basic Books.

Lieberman, H., Paternò, F., & Wulf, V. (Eds.). (2006). *End user development*. Dordrecht: Springer.

Lovins, A. B. (1977). *Soft energy paths: Toward a durable peace*. San Francisco: Friends of the Earth.

Martin, M., & Schinzinger, R. (1983). *Ethics in engineering*. New York: McGraw Hill.

Merton, R. (1942). Science and technology in a democratic order. *Journal of Legal and Political Sociology, 1*(1945), 115–126. Reprinted as "The normative structure of science", in R. K.

Merton. In N. W. Storer (Ed.), *The sociology of science: Theoretical and empirical investigations*. Chicago: University of Chicago Press, 1973.

Mitcham, C. (1994). Engineering design research and social responsibility. In K. S. Shrader-Frechette (Ed.), *Ethics of scientific research* (pp. 153–168). Lanham, MD: Rowman and Littlefield.

Mitcham, C. (1999). Why the public should participate in technical decision making. In R. von Schomberg (Ed.), *Democratising technology: Theory and practice of a deliberative technology policy* (pp. 39–50). Hengelo: International Centre for Human and Public Affairs.

Norman, D., & Draper, S. (Eds.). (1986). *User-centered system design: New perspectives on human–computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Nozick, R. (1974). *Anarchy, State, Utopia*. Cambridge, MA: Harvard University Press.

Ortega y Gasset, J. (1939). *Meditación de la técnica*. In *Obras completas* (1st ed., Vol. 5, pp. 317–375). Madrid: Revista de Occidente.

Polanyi, K. (1944). *The great transformation*. New York: Farrar and Rinehart.

Popper, K. (1945). *The open society and its enemies*. London: Routledge. Revised and expanded editions: rev. (1950), 2nd rev. (1952), 3rd ed. rev. (1957), 4th ed. rev. (1962), and 5th ed. (1966).

Raymond, E. S. (1999). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Beijing: O'Reilly.

Sachs, W. (Ed.). (1992). *The development dictionary: A guide to knowledge as power*. London: Zed Books.

Schumacher, E. F. (1973). *Small is beautiful: Economics as if people mattered*. New York: Harper and Row.

Scrivener, S. A. R., Ball, L. J., & Woodcock, A. (Eds.). (2000). *Collaborative design: Proceedings of codesigning 2000*. London: Springer.

Seffah, A., Gulliksen, J., & Desmarais, M. C. (Eds.). (2005). *Human-centered software engineering: integrating usability in the software development lifecycle*. Dordrecht: Springer.

Torvalds, L. (1992). LINUX's history. July 31, 1992. A collection of emails with narrative notes available on a number of web sites. Accessed from www.cs.cms.edu/~awb/linux.history.html Feb 28, 2007.

Torvalds, L., & Diamond, D. (2001). *Just for fun: The story of an accidental revolutionary*. New York: HarperCollins.

Turner, F. (2006). *From counterculture to cyberworld: Stewart brand, the whole earth network, and the rise of digital utopianism*. Chicago: University of Chicago Press.

Verbeek, P.-P. (2005). *What things do: Philosophical reflections on technology, agency, and design*. University Park, PA: Penn State University Press.

von Bertalanffy, L. (1968). *General system theory: Foundations, development, applications*. New York: Brazillier.

Weber, S. (2004). *The success of open source*. Cambridge, MA: Harvard University Press.

Whitehead, A. N. (1925). *Science and the modern world*. New York: Macmillan.

Winner, L. (1980). Do artifacts have politics? In *Daedalus* (no. 1, Vol. 109, pp. 121–136) (Winter 1980). Reprinted in Winner, L. (1986). *The whale and the reactor: A search for limits in an age of high technology* (pp. 19–39). Chicago: University of Chicago Press.