NOTES ON THE ECONOMIC HISTORY OF LINUX (APPENDIX TO R&D OUTSOURCING)

SALVATORE MODICA*

Some information is collected here about the economic history of Linux, in the hope that some readers will find it useful; it is referred to in [13]. After mentioning the legal regime under which the system works we shall briefly go into past history and current situation. Some comments on what we may abstract from the open source experience are also included.

Legal Underpinning: the GPL. The legal twist that gave birth, in the late eighties, to Open Source software is the *General Public Licence*, whose original idea is owed to the MIT programmer Richard Stallman. As all licences, the GPL and the several less 'radical' variants which are around by now are written by lawyers and for lawyers; we shall pass on what we understand about them. [1]

Open source means that the user must be able to 'see the source [code]', but there is more in the GPL; the essential twist is, in Stallman's original wording, to "turn copiright into copyleft": whereas copyright contains restrictions to use, modify and distribute a product, copyleft contains the restriction to restrict those things. [2] It is interesting that the GPL does not restrict the right to *sell* the programs covered by the licence —there is no need to. As Stallman puts it (`www.gnu.org/philosophy/selling.html`),

> "[there are] no requirements about how much you can charge for distributing a copy of free software. You can charge nothing, a penny, a dollar, or a billion dollars. It's up to you, and the marketplace, so don't complain to us if nobody wants to pay a billion dollars for a copy."

Indeed nobody will, because the impossibility to restrict redistribution induces competition between buyer and seller: buyer cannot make money by reselling product because if it tries to sell it for p the original seller can sell it for $p - \epsilon$; hence the buyer will not pay more than redistribution cost, so that the original seller will not make money in the first place.

Thus the rule underlying open source software production is in essence that every user (potentially user/developer) has the right to see what the users before her have done, and must pass this right on to the subsequent users. Of course, thinking of Linux again, this is surely not sufficient to generate a process of such

* November 2005 Financial Support: MIUR. Affiliation: Dip. Scienze Economiche Finanziarie e Aziendali, Università di Palermo, Viale delle Scienze, 90128 Palermo, Italy. email `modica@unipa.it`.

[1] The two main sites about the topic are `www.gnu.org` and `www.opensource.org`, and contain all licences and extensive discussions. Useful for general understanding is in particular the commonly accepted Open Source *Definition*, which contains the essential requirements which open source licenses should satisfy. The idea of a definition which would serve as basis for the licences originates with Bruce Perens; his original version is in the Articles section of his web site, `www.perens.com`, of independent interest.

[2] "The central idea of copyleft is that we give everyone permission to run the program, copy the program, modify the program, and distribute modified versions —but not permission to add restrictions of their own. Thus, the crucial freedoms that define 'free software' are guaranteed to everyone who has a copy; they become inalienable rights." Stallman, at `www.gnu.org/gnu/thegnuproject.html`.

import; there are motivational issues (why do programmers contribute to an open source project if they cannot 'make money'?), and problems of coordination and technical feasibility in the way; we now turn to these.

Genesis of the project. We touch upon three points: *(i)* actors' motivations, *(ii)* organization, and *(iii)* technology of the Linux project.

(i) Motivations. About his decision to make Linux freely available, his creator Linus Torvalds says "[it] wasn't some agonizing decision that I took from thinking long and hard on it: it was a natural decision within the community that I felt I wanted to be a part of." (interview in First Monday, [4]). That crucial decision was effectively the only one available to him at the time, as Linus himself declares in his book [17] (chapter 2, section IX); but the point remains that it was a natural choice. For that community he wanted to be a part of was a community to which all members were feeling good to belong, in a somehow deep sense. It was like when hippies liked to be hippies in the States, or the '68–guys liked to be what they were in Europe; and more and more all those involved perceived to be part of something great and important (at least this is the impression I get by watching the hackers' community from outside). Viewed in this light the economics question "How come all these guys have contributed apparently for free" sounds stupid at first sight; but of course it must be read as "What is there *besides* social magic behind their motivations". [3] First there is some 'individual' magic, like there is in mathematics; Torvalds, for one, thinks this —'the fun'— is the main thing (Torvalds [17]). [4] Then there are two purely economical forces: the signalling motive, and the direct use–value to the user–developer of his own contribution. On the first, the insiders' view is not difficult to guess; see e.g. Raymonds [15], or Torvalds in [4]; critical is also Benkler [1]. [5] User value on the other hand has certainly been decisive for the major open source projects (see e.g. Lerner–Tirole [11]). And it is worth noting that even outside the software industry, the relevance of user–driven product development is widely recognized; see most notably von Hippel [18, 19, 20]. Also, to the individual use value one must add the cumulative effect of concurring contributions —as Ganesh Prasad puts it for software development, "Each programmer contributes a brick and each gets back a complete house in return." [6] This picture of motivations is recalled when we comment on replicability of the open source model outside software production in the concluding section of the text.

---

[3] Concerning 'who contributed what', in their survey on 13,000 contributors to open source projects Ghosh and Prakash [5] found that three quarters made only one contribution, but at the other end nearly three quarters of contributions came from the top ten percent of contributors. I suppose the social excitement factor alone is enough to explain the one–timers' contributions; the question is really about the hard–working guys. Note that even for the latter the social factor is not irrelevant, for they were leaders of a large generational movement, and obtaining and maintaining such a position may well be worth a lot of hard work.

[4] In fact 'fun' is more for him: it is the third and last of the three stages of evolution of humanity according to his (non–trivial) theory of evolution, the first two being survival and social order; see [17].

[5] Torvalds, after much pressure from the interviewer responds "Yes, there are issues involved with 'getting value back' from your involvement [...] but the first consideration for anybody should really be whether you'd like to do it *even if* you got nothing at all back". Benkler adduces the fact that some of the most important projects, like the Apache Web Server and the Free Software Foudation, do not provide personal attribution to the code they produce. In fact much more is true, cfr. Ghosh–David [6]: in the Linux kernel consistently more than half of the code is unsigned; and those packages whose lines of code are entirely signed constitutes the 0.66% of total kernel packages.

[6] `http://linuxtoday.com/infrastructure/20010412006200PBZCY--`.

(ii) Organization. [7] The hierarchical organization of the Linux project (and of most of its satellite projects) is usually of a 'benevolent dictatorship' (cfr. Dafermos [3]); the dictator must also be *trusted*, and trust is conferred by public legitimacy. As to productive organization, the clearest insight for understanding emergence of 'peer production' comes in my opinion from Benkler [1] (NYU School of Law). Benkler makes a conditional statement, *given* strong enough actors' motivations (cfr. above) and technological feasibility (on this shortly). With these assumptions in place, to the two dimensions of transaction costs/organization costs responsible for the firm/market tradeoff in Coase's theory Benkler adds a third dimension: that of 'information opportunity costs', and correspondingly a third alternative mode of production: peer production. The idea is that the latter may prevail due to the advantages which decentralized information gathering and exchange gives in identifying and allocating creative work to the more appropriate jobs.

(iii) Technology. The essential characteristics of the process are well understood: there is a substantial initial 'core' of potential widespread use (Weber [21]); the subsequent product development is modular (also Benkler [1] and Lerner–Tirole [11]); the size of the modules is small ([1], [11]; Benkler uses the term 'granularity'); and modules integration (quality control and decision processes) is managed effectively (cfr. [1]). Lerner–Tirole argue that lack of granularity is the main technological obstacle to transposition of the open source/peer production model to other industries: "[. . .] In many industries, development of individual components require large team work and substantial capital costs" ([11], p. 231). This is undoubtedly true, but not necessarily pervasive; von Hippel [20] for instance cites old empirical research on technological innovation (concerning the post–war decades) showing that both in Rayon manufacture and computer hardware the cumulation of a multitude of minor technical changes is "responsible for much or most technical progress" ([20] p.14). Distinguishing the different stages of product maturity, is seems reasonable to expect large teams and investments more frequently needed in an initial phase, followed by a cumulation of minor improvements taking place in a subsequent one.

The Present. In the last couple of years much has changed. Ghosh and Prakash [5] found in 2000 that 75% of contributors to open source projects were one–timers; there are no surveys about the current situation yet, but the obvious guess is that the times of one–timers have gone (just read on). We will talk separately about *(i)* for–profit firms commercializing Linux, and *(ii)* the *Open Source Development Labs*.

(i) Red Hat & C. In a paper appeared in August 2003, Haruvy et al. [7] solved what would have been Red Hat's optimal control problem with commercializing open source software in a situation like the one of the 2000 Survey, namely: if such a firm charges a short–run profits maximizing price, the high profit realized may induce spite in the hackers' community, hence decreased contributions, hence lower future product quality, hence ultimately lower profits. Well, forget it. Remember the Red Hat Box selling at around 50 US Dollars? In mid–October 2003 it was still selling, but it was pretty hard to find in their web site (at least for me); by end–October, it had disappeared completely! The point is that it is not what

[7] In the sense of *industrial* organization. Many have written about the organization of the project from a *social* point of view. Not surprisingly the parallels to the academic model of open knowledge production are ubiquituos: besides Raymond [15] see e.g. Himanen [8], who starts from Plato's Academia, and Benkler [1] who is himself an academic. From an evolutionary point of view, interesting is the essay by Kuwabara [10] who, based on the gift–culture idea of Raymond, gives an interpretation of the process in the light of the the Santa–Fe approach to the dynamics of complex systems. A link to the work of J.B. Arthur from Santa–Fe comes also from Dafermos [3] in connection with increasing returns.

they sell any more; they sell Red Hat 'Architecture', and Red Hat 'Solutions', to firms who become Red Hat customers. No role for hackers left to play. Open Source, Linux–based software is now getting ready to replace proprietary (mostly Unix–based) software infrastructure at mission–critical level, in the communication market. The operating system is just a part of a much more complex product, and competition is growing between 'Red Hat Architecture' against 'IBM Linux Solutions', 'SuSe for the Enterprise', etc. (incidentally Red Hat and VA Software, the two leading firms of the sector quoted at Wall Street, have seen their stock value more than doubled in the last three months, August–October 2003).

(ii) The OSDL. What about Linus, who started it all? Miniaturized along with old kernel problems? Contrarily to what one could expect, the answer is no — indeed, all the opposite. To migrate to Linux in corporate data centers and in telecommunications networks, the interested companies want reassurance that the system meet some critical 'carrier-grade' requirements. And since different firms have different technologies and priorities, sofware developers on their part need to know what exactly these requirements are, and how they are ranked in terms of priority. So the crucial step becomes the creation of a 'focal' set of requirements *definitions* and priority ranking (of course evolving with time), based on inputs from the business sector and to which developers can refer for their programming objectives. Supported by a global consortium of Information Technology industry leaders, the *Open Source Development Labs*, a non-profit organization, was founded in 2000 for exactly that purpose. [8] Most big corporations were involved, but the vital 'authority', in the sense of recognition from the community the way Torvalds had been for the early kernel development, was missing. What exactly was missing is easily guessed: Linus in person, of course. Well, since June 2003 that is where he is: in this new Linux world, again at what is becoming an important gravitational center of it. In essence, in the market sketched sub *(i)* above, where product complexity makes the source more and more 'hidden', the role if OSDL is that of keeping the core of it common and open.

Comments on Open Source and Patent Policy

We now tentatively spell out the conditions under which an open source model might be applied outside software production. A final point about growth policy concludes.

Core and Trusted Authority. The fact that it is essential that there be a 'substantive initial core' which has the potential to become of widespread use is well recognized (cfr. e.g. Benkler [1], Weber [21]). And from the organizational point of view I would like to stress the importance of a 'central authority' (like Linus in person at the beginning of the Linux project and the OSDL these days).

Product Cycle and Quality Circles. To peer–develop an initial product each contributor must obvously have a higher payoff from revealing than from concealing his work and using it only for himself. Assuming that it is feasible to emarginate those who conceal from sharing subsequent improvements, the requirement is that there be an equilibrium in which each reveals her contribution to the community

[8] The web site is `www.osdl.org`. The consortium includes from hardware producers like Cisco, Dell, HP, IBM, Intel, Mitsubishi, Nec, Sun and Toshiba, to firms involved in telecommunications like Ericsson and Nokia, and more software oriented firms like Linuxcare, Red Hat, SuSe, Turbolinux. The directory `lab_activities/carrier_grade_linux/documents.html` contains the main charter document, together with a 'Technical Scope White Paper'; the analogous White Paper for Data Centers is in `lab_activities/data_center_linux/documents.html`. The technical Requirements Definition (version 2.0) is of course also available at their site. A 'need to know' paper about carrier grade linux, written for engineers, is Mehaffey [12].

and benefits from the others' ones (the "each contributing a brick to have a complete house in return" of Ganesh Prasad, cfr. p. 2 below). The conditions for this seem to be most favourable: (i) for complex products, where improvements occur in all directions/parts, (ii) in the initial phase of development, where due to decreasing returns to research the value of individuals' contributions is highest. We speak of 'directions' of improvements here because for complex products a more appropriate visualization than the traditional 'quality ladder' (cfr. Scotchmer [16]) seems to be an image of quality (lager and lager) 'circles'.

User Value and Granularity. On the motivational side one surely cannot rely on social excitement (cfr. Appendix B), but the same must be said of the often quoted signalling motive (see e.g. Lerner–Tirole [11]): as reported in the Appendix, after close inspection it is found rather weak even in the software case. What remains is the user–value, which as we learn from the work of von Hippel ([18]–[20]) should not be undervalued. It is to be stressed that one should have in mind here the firm–user more than the consumer–user: e.g. a firm innovating a process machine which it uses in a major product line may make substantial profits from the innovation; or, think of brakes improvements on the part of a producer of racing cars or aircrafts. Thus, 'large team work' ('lack of granularity') is not necessarily an insurmountable barrier. All the more so given that in a peer production process cooperation has no complicated property–related drawbacks; see, most notably, the current experience of the OSDL (see again Appendix B) where all major competitors in the telecommunication market are involved.

Intermediate Products. The discussion so far points to a specific class of goods: the complex intermediate products. A moment's thought suggests the qualification that they should not be crucial for gaining competitive edge; for example, brakes but not components/solutions affecting fuel consumption for family cars in Europe (where fuel is highly taxed and fuel consumption is one of the first things consumers watch and car makers advertise).

Timing. The Linux kernel has develped so fastly because 'the world' was just ready to take up Linus' work and improve upon it. Quite likely, if Linus had written his version 0.01 in 1975 instead of 1991, history would have been very different. In fact we have an example of something like this happening: John von Neumann's insights on computers' architecture date 1945, but it took about ten years before they influenced industrial production (which they did pervasively when time was ripe; cfr. Mowery–Rosenberg [14]). The point here is obviuos but it may be crucial: for peer production, peers must be there and ready.

Policy and Initial Core. This is not a paper about intellectual property and desirability of patents, and we will not raise the point in the last paragraph. See Bessen–Maskin [2] and Scotchmer [16] for 'problems' with patents in the presence of cumulative innovation. Remaining focused on peer production, we remark that the weak spot in such processes which the preceeding observations point to is the existence of the 'substantive initial core' —there are not many Torvalds or von Neumann around.

Given an initial core, under the conditions listed above product development is possibly faster in a patent–free, open source environment rather than in a proprietary system; but as we all know absence of patents may deter production of primary innovations/initial cores. Thus the trade-off for growth which seems to emerge for patent policy is between having more primary innovations with slower improvement against having fewer innovations with more extensive development.

The additional policy dimension may be the public funding and/or acquisition of primary innovations to be released with open source (that is under some kind

of GPL) whenever this seems propitious; the above considerations are intended to contribute to identify conditions under which it may be so.

References

[1] Benkler, Yochai (2002): Coase's Penguin, or, Linux and 'The Nature of the Firm', *Yale Law Journal* **112**

[2] Bessen, James and Eric Maskin (2000): Sequential Innovation, Patents and Imitation, MIT Working Paper no. 00-01

[3] Dafermos, George N. (2001): Management and Virtual Decentralised Networks: The Linux Project, *First Monday* **6** No.11

[4] First Monday (1998): Interview with Linus Torvalds: What motivates free software developers? *First Monday* **3** No.3

[5] Ghosh, Rishab Aiyer and Prakash, V.V. (2000): The Oribiten Free Software Survey, 1st Edition, online at `http://orbiten.org`; also reported in *First Monday* **5** No. 7 (July 2000).

[6] Ghosh, Rishab Aiyer and Paul A. David (2003): The nature and composition of the Linux kernel developer community: a dynamic analysis, *Stanford Project on Economics of Open Source Software*, mimeo Stanford University

[7] Haruvy, E., A. Prasad and S.P. Sethi (2003): Harvesting Altruism in Open–Source Software Development, *Journal of Optimization Theory and Applications* **118**: 381-416

[8] Himanen, Pekka (2001): *The Hacker Ethic and the Spirit of the Information Age*, Random House

[9] Kultti K. and T. Takalo (2000): Incomplete contracting in an R&D project: the Micronas case, *R & D Management* **30**: 67–77

[10] Kuwabara, Ko (2000): Linux: a Bazaar at the Edge of Chaos, *First Monday* **5** No.3

[11] Lerner, Josh and Jean Tirole (2002): Some Simple Economics of Open Source, *Journal of Industrial Economics* **50**: 197–234

[12] Mehaffey, John (2002): Carrier Grade Linux: What You Need to Know, *Communication Systems Design*, online at `www.commsdesign.com/story/OEG20020827S0008`

[13] Modica, Salvatore (2006): "Knowledge Transfer in R&D Outsourcing: an Incentive-Constrained view", mimeo

[14] Mowery, David C. and Nathan Rosenberg (2000): *Paths of Innovation: Technological Change in 20th–Century America*, Cambridge University Press

[15] Raymond, Eric S. (1998): Homesteading the Noosphere, *First Monday*, **3** No.10

[16] Scotchmer, Suzanne (1999): Cumulative Innovation in Theory and Practice, GSPP Working Paper no. 240, UC Berkeley

[17] Torvalds, Linus and David Diamond (2001): *Just for Fun: The Story of an Accidental Revolutionary*, Harper Business

[18] von Hippel, Eric (1986): Lead Users: a Source of Novel Product Concepts, *Management Science* **32**: 791–805

[19] von Hippel, Eric (1998): Economics of Product Development by Users: Impact of 'Sticky' Local Information, *Management Science* **44**: 629–644

[20] von Hippel, Eric (2002): Horizontal Innovation Networks —by and for Users, *MIT Sloan School of Management* WP No. 4366–02

[21] Weber, Steven (2000): The Political Economy of Open Source Software, BRIE Working Paper 140, Berkeley University