Open Source Software and the Economics of Organization

Giampaolo Garzarelli
Università degli Studi di Roma, 'La Sapienza'
Dipartimento di Teoria Economica e
Metodi Quantitativi per le Scelte Politiche
Piazzale Aldo Moro, 5
00185, Rome, Italy
+39.06.49910627 (phone)
+39.06.4453870 (fax)
giampygarz@hotmail.com

Final Draft: April 30th, 2002

Forthcoming in a book on the new economy edited by J. Birner and P. Garrouste published by Routledge.

***Abstract**:* Open source software development has organizational characteristics that are out of the ordinary (*e.g.*, no hierarchy, self-organization, self-regulation, and no ownership structure). The study suggests that this organization of work can be explained by combining the recently developed organizational theory of professions with the classic one of clubs. Still, the *explanans* falls within the broad rubric of the knowledge approach. The claim is in fact that this organization is at least as good as a firm in sharing rich types of information in real time because (*a*) constituents have symmetry of absorptive capacity, and (*b*) software itself is a capital structure embodying knowledge. Indeed, in this regard the study goes so far as to suggest that the distinction between input (knowledge) and output (software) is somewhat amorphous because knowledge and software are not only the common (spontaneous) standards, but also the nonrivalrous network products being shared.

***JEL-classification**:* D2, H0, K0, L0, L2, L5, O3.

Keywords: software, open source software, economics of organization, economics of professions, clubs, technological clubs, absorptive capacity, information technology, Internet, computers, networks, innovation, standards, new economy.

Acknowledgements: I thank Cristiano Antonelli, Metin M. Coşgel, Steven R. Cunningham, George N. Dafermos, Roger Koppl, Jorge L. Laboy-Bruno, Yasmina Reem Limam, and Mark S. Miller. My intellectual debt to Richard N. Langlois is beyond the few references to his work. The paper was written while I was at the Economics Department of the University of Connecticut, Storrs, USA.

> … the productivity of social cooperation surpasses in every respect the sum total of the production of isolated individuals.
>
> (Mises 1981[1933]: 43)

> It should be noted that most inventions will change both the costs of organizing and the costs of using the price mechanism. In such cases, whether the invention tends to make firms larger or smaller will depend on the relative effect on these two sets of costs. For instance, if the telephone reduces the costs of using the price mechanism more than it reduces the costs of organizing, then it will have the effect of reducing the size of the firm.
>
> (Coase 1937: 397, note 3)

INTRODUCTION

It is often remarked that innovation in computer technology is profoundly affecting the organization of production and of consumption of contemporary society. For instance, consumers are said to be increasingly participant in the production process, leading to an increase in the modular nature of most products and organizations, and to an increase in the thickness of most markets (e.g., Cox and Alm 1998; Dolan and Meredith 2001).

But it is seldom acknowledged that there is a complementary, and at least equally important, aspect of this technological innovation: its software counterpart (some exceptions are Baetjer 1998 and Lavoie ch. ? this volume). This work attempts to do some justice to this shortcoming by describing some elements of a new type of organization of work, the one generated by open source software development.

Open source includes such software success stories as Apache, Perl, Sendmail and Linux. To give but a few recent statistics on the phenomenon — although these figures are subject to frequent fluctuations — as of March 2002 the top web server is Apache, with 53.76 per cent of the market, for a total of 64.37 per cent of all active sites (see Tables 1 and 2). And according to the Linux Counter, as of April 30, 2002 there are 125,549 registered users and 95,834 registered machines (<http://counter.li.org/>); but the Counter also estimates the worldwide Linux users to be 18 million (<http://counter.li.org/estimates.php>). It is also interesting to notice that some of the most successful PC makers have recently also become some of the top Linux server vendors (see Table 3).[1]

---

1 To be more precise, 'Linux' refers to the kernel of the operating system; while the entire project is called 'GNU/Linux' because it contains parts of the GNU project, started by Richard Stallman in 1984. See e.g. the chapter by Stallman in DiBona *et al.* (1999: 53-70).

The influence of open source both as a business and as a software development model have been vast (e.g., *Release 1.0* 1998*:* 8ff; *passim* DiBona *et al.* 1999*; passim* Rosenberg 2000). For example, Netscape decided to develop an open source browser, Mozilla, in 1998; IBM adopted Apache as a web server for its Websphere product line; while Apple ships Apache along with their operating system. And Microsoft, seen by many open sourcers as the ultimate enemy,[2] is looking into the possibility of going open source in some products (perhaps because of the pressures originating from the ongoing antitrust litigation) by launching so-called shared source: seemingly, a hybrid of proprietary and open software (D. H. Brown Associates, Inc. 2001).[3]

The open source philosophy assures a 'self-correcting spontaneous' organization of work that is 'more elaborate and efficient than any amount of central planning could have achieved' (Raymond 2001: 52). By drawing on the recently developed organizational theory of professions and on the classic theory of clubs, the pages that follow will attempt to describe how this organization can exist. To this end, it is first of all (and primarily) suggested that the organizational economics of open source software development is so complex that a theory that has the ambition to explain it needs to begin by looking at the nature of the *knowledge* involved in the production and consumption of open source software itself.[4]

The main advantage of following such a cognitive approach is that it lends itself well to explaining the self-organizing as well as the self-regulating properties of open source economic organization. That is to say that the approach gives solid foundations to the eclectic organizational theory that this exploratory essay proposes.

---

2 See for example cWare (nd); but compare Eunice (1998).

3 For Microsoft's reaction to the open source phenomenon see the 'Halloween Documents' (some internal Microsoft memoranda that try to assess competition from open source that later became public): <http://www.opensource.org/halloween> (accessed 10 February 2000).

4 Take note that by 'consumption of open source' I refer to consumption on the supply-side: consumption by open source producers. I do not consider, in other words, 'downstream' consumption of open source, the one made by individuals using open source who are not at the same time involved in its production; even if these downstream consumers may suggest to the open source community about how to improve software.

Table 1: Top Developers

Developer	February 2002	Percent	March 2002	Percent	Change
Apache	22,462,777	58.43	20,492,088	53.76	-4.67
Microsoft	11,198,727	29.13	12,968,860	34.02	4.89
iPlanet	1,123,701	2.92	889,857	2.33	-0.59
Zeus	837,968	2.18	855,103	2.24	0.06

Note: iPlanet is the sum of sites running iPlanet-Enterprise, Netscape-Enterprise, Netscape-FastTrack, Netscape-Commerce, Netscape-Communications, Netsite-Commerce & Netsite-Communications. Microsoft is the sum of sites running Microsoft-Internet-Information-Server, Microsoft-IIS, Microsoft-IIS-W, Microsoft-PWS-95, & Microsoft-PWS.

Source: Netcraft (<http://www.netcraft.com/survey/>).

Table 2: Active Sites

Developer	February 2002	Percent	March 2002	Percent	Change
Apache	10,147,402	65.18	9,522,954	64.37	-0.81
Microsoft	4,069,193	26.14	3,966,743	26.81	0.67
iPlanet	283,112	1.82	265,826	1.80	-0.02
Zeus	177,225	1.14	170,023	1.15	0.01

Note: iPlanet is the sum of sites running iPlanet-Enterprise, Netscape-Enterprise, Netscape-FastTrack, Netscape-Commerce, Netscape-Communications, Netsite-Commerce & Netsite-Communications. Microsoft is the sum of sites running Microsoft-Internet-Information-Server, Microsoft-IIS, Microsoft-IIS-W, Microsoft-PWS-95, & Microsoft-PWS.

Source: Netcraft (<http://www.netcraft.com/survey/>).

Table 3: Top Linux Server Vendors

Vendor	Market Share
Compaq	25%
IBM	10%
HP	7%
Dell	7%
Fujitsu Siemens	3%
Others	48%

Source: IDC, 2000, Q4, 1999 unit shipments, cited in West and Dedrick (2001: Table 2).

Perhaps the most interesting point that shall emerge is that this 'atypical' organization is at least as good as a firm in sharing rich types of information in real time.[5] I submit that the two reasons for why this is so are (*a*) that constituents have symmetry of absorptive capacity, and (*b*) that software itself is a capital structure embodying knowledge. Indeed, in this regard, I go so far as to suggest that the distinction between input (knowledge) and output (software) is in some ways amorphous because knowledge and software are not only the common (spontaneous) standards, but also the nonrivalrous network products being shared.

CONTEXTUALIZATION

In general, software development is very complicated and involves a substantial amount of experimentation and trial-and-error learning. This renders it a cumulative process where improvements are incremental rather than radical. Contrary to, e.g., pharmaceuticals, innovation is not discrete and monolithic, but often builds on previous software. In addition, innovation usually proceeds at a faster pace than most other industries because numerous individuals simultaneously try multiple approaches to solve the same problem. Clearly, such process is imbued with uncertainty. But the multiple approaches and the numerous individuals also create a great variety of potential improvements, arguably more than any single individual, thus increasing the possibilities for success. In turn, the variety leads to new problems and to new trial-and-error learning.

5 'Real time' in the computer science sense of being able to do, evaluate or react to things as they are happening, without (much) delay. Two classic examples of real time behavior are a telephone conversation and software that tries to constantly track weather conditions to attempt to offer forecasts. For an organizational application of this notion see Langlois and Robertson (1995: ch. 3).

What renders all this possible and at the same time makes software so supple is its peculiar nature, namely, its modularity. Modularity is one method to manage complexity. Programs, especially more modern ones of the object-oriented type, are *per se* composed of different, interacting modules; and it is possible to change a part of a module or an entire module without knowing all information about the program that the module belongs to and without altering other modules or the overall purpose of the program (Baetjer 1998). This is possible because through modularization a program hides information among modules while at the same time allowing for their communication — this principle is known as *information hiding*.

Originally introduced by Parnas (1972), information hiding assures that software is extendible, compatible and reusable.[6] According to this principle in fact 'system details that are likely to change independently should be the secrets of separate modules; the only assumptions that should appear in the interfaces between modules are those that are considered unlikely to change' (Parnas *et al.* 1985: 260). Consequently, information hiding stimulates the division and specialization of knowledge, allowing productive knowledge to converge to its most valued use. And all this entails that the only benchmark to assess the 'efficiency' of a particular software is not so much its ability to perform its tasks as its ability to evolve in order to potentially perform its tasks even better (Baetjer 1998).

The traditional, corporate approach to software development is centred on hierarchical relations. The decision of what software to develop, test or improve comes from the top of the hierarchy. Open source software development, in contrast, is practically based on the absence of hierarchy.[7]

But as others have pointed out, this does not at the same time necessarily imply that all open source software projects lack a sometimes even rigid organizational structure.[8] Apparently, it is not rare for an open source project to be terminated in the absence of a meritocratic management structure organizing the development process. Conversely, it is also apparently not rare for a very interesting project to fail to create momentum because of organizational rigidities.

What exactly is open source software then? Tim O'Reilly, founder and CEO of O'Reilly & Associates, a company that publishes many books on open source, offers a concise definition.

---

6 But compare Brooks (1975: 78ff.).

7 In some cases there still is some authority, however. In the case of Linux, for example, Linus Torvalds (or a close collaborator) decides which software code to accept into the Linux kernel.

8 Notably George N. Dafermos in private communication with the author.

> Open source is a term that has recently gained currency as a way to describe the tradition of open standards, shared source code, and collaborative development behind software such as the Linux and FreeBSD operating systems, the Apache Web server, the Perl, Tcl, and Python languages, and much of the Internet infrastructure, including Bind (The Berkley Internet Name Daemon servers that run the Domain Name System), the Sendmail mail server, and many other programs. … [But] open source (which is a trademark of the Open Source Initiative – see <http://www.opensource.org>), means more than the source code is available. The source must be available for redistribution without restriction and without charge, and the license must permit the creation of modifications and derivative works, and must allow those derivatives to be redistributed under the same terms as the original work.
>
> (O'Reilly 1999: 33-4, emphasis removed)

Notably, the participation to open source projects is voluntary (there's strong self-selection) and supervision is assured on a peer review basis. [9]

The origins of open source go back to the so-called hacker culture.[10] Hackers are very creative software developers who believe in the unconditional sharing of software code and in mutual help. The advent of the microcomputer diffused this ethos beyond the narrow confines of the academic environments where it originally developed (MIT, Stanford, and Carnegie-Mellon), and it multiplied digital linkages. In effect, it dematerialized the need for concentration of hackers in specific laboratories, moving their concentration to cyberspace.

Eric Raymond, hacker and author of the very influential open source 'manifesto' *The Cathedral and the Bazaar* (2001), summarizes the fundamental philosophy underlying the open source community in the context of his discussion of Linux.

> … Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.
>
> Or, less formally, "Given enough eyeballs, all bugs are shallow". I dub this: 'Linus's Law'. …

9 There are several licenses governing open source. Analyzing these in detail necessitates a study of its own. See especially DiBona *et al.* (1999: Appendix B), Rosenberg (2000: chs 6, 7 and Appendix A) and Raymond (2001: 73ff.).

10 See in particular Raymond (2001: 1-17; 169-91).

> In Linus's Law … lies the core difference underlying the cathedral-builder and bazaar styles. In the cathedral-builder view of programming, bugs and development problems are tricky, insidious, deep phenomena. It takes months of scrutiny by a dedicated few to develop confidence that you've winkled them all out. Thus the long release intervals, and the inevitable disappointment when long-awaited releases are not perfect.
>
> In the bazaar view, on the other hand, you assume that bugs are generally shallow phenomena — or, at least, that they turn shallow pretty quickly when exposed to a thousand eager co-developers pounding on every single new release. Accordingly you release often in order to get more corrections, and as a beneficial side effect you have less to lose if an occasional botch gets out the door.
>
> (Raymond 2001: 30-1, emphasis removed)

Let's try to identify some of the necessary ingredients for an organizational theory of bazaar-style software development.

THE ATTRIBUTES OF PROFESSIONS

Deborah Savage, in an innovative piece, proposes the following economic definition of a profession: a 'profession is a network of strategic alliances across ownership boundaries among practitioners who share a core competence' (Savage 1994: 131). The keyword here is *competence*. As the literature beginning with the resuscitated contributions by Penrose (1995[1959]) and Richardson (1998: ch. 10) has made clear, in fact, production is not as simple as making a soup: it is not so much a question of putting some inputs together (K,L), performing some manipulations $f(\bullet)$, and, *voilà*, obtaining some output X.[11] Rather, it is a complex process involving abilities, experience, and learning — it is, to put it in more general terms, a cognitive-based process encapsulating different routines and capabilities evolving through time (Nelson and Winter 1982).[12]

11 The image is Leijonhufvud's (1986: 203).

12 To clarify, next to competence, the literature also speaks of 'routines' (Nelson and Winter 1982), 'capabilities' or 'dynamic capabilities' (Langlois and Robertson 1995). *Stricto sensu*, routines are what an organization does, they are the economic equivalent of the biological genes or economic memory; capabilities/dynamic capabilities are what an organization can do, e.g. if circumstances change and redeployment of resources takes place — they are directly complementary to competencies; competencies are the core abilities that an organization possesses, i.e., what an organization specializes in depends on its competence (although, in time, competencies may change). To schematize: *routines* $\in$ *capabilities* $\in$ *competencies*. Yet, these categories are not mutually exclusive as the (illustrative) classification might suggest; in fact, all the notions are quite slippery.

The capabilities involved in producing goods and services are often based on tacit knowledge in the sense of Michael Polanyi (e.g., 1966).[13] In the specific case of professions, we have a knowledge that is highly tacit and specialized or, to use a catchall wording, we have 'esoteric knowledge' (Savage 1994: 135-6). This knowledge represents the elemental component of professions. For example, it manages to couple competencies, to coordinate, and so on. It therefore offers the rationale for the existence of professions, and it provides for their cohesion and coherence — in a way, for their boundaries as well. In brief then, professional capabilities are a form of capital representing the productive essence of the network, and more generally coevolving with the network itself.

To change viewpoint on the matter, absent esoteric knowledge, professionals and, *a fortiori*, their coupling would not exist. We apparently face a situation where the division of knowledge (Hayek 1948: ch. 2) drives the division of labour. Professions — like most other organizational forms — then exist for epistemic reasons or, what boils down to the same thing, for Hayekian (qualitative) coordination, that is, for coordination beyond mere price and quantity (compare Malmgren, 1961).[14]

An important corollary is that the fairly symmetric nature of capabilities present in professions assures that the 'absorptive capacity' (Cohen and Levinthal 1990) — i.e., some similar capabilities — is virtually always present.[15] Indeed, it is the spontaneous orchestration of knowledge generated by the symmetry of absorptive capacity that creates a profession's complex self-organization — with, notably, absence of ownership structure[16] — as well as its external economies, such as uncertainty reduction, mutual monitoring, incentive alignment, trust and, most important for our discussion (as we shall see shortly), reputation.

These characteristics do not necessarily mean, however, that each professional has exactly the same capabilities; otherwise we would not be in the presence of complex self-organization. Rather, the point is that the capabilities share a rudimentary (esoteric) knowledge base – the core competence noted above – that affords the absorptive capacity, a necessary condition for spontaneous self-organization.

---

13 Polanyi's tacit dimension that is opposite the explicit one, is akin, in many ways, to Ryle's (1971[1946]) dichotomy between 'knowledge that' (explicit) and 'knowledge how' (tacit); the distinction made by de Solla Price (1965) between technological (how) and scientific (why) knowledge is also relevant here.

14 A point of view, incidentally, compatible with Coase's (1937) original story; see for example Langlois and Robertson (1995) and Garzarelli (2001).

15 Although not widely remarked upon (an exception is Langlois and Robertson [1995]), this is, in effect, the flip-side of competencies, at least in normal periods of production and exchange, *i.e.*, those involving little radical innovation.

16 *Contra* Hansmann (1996).

In sum, the general organizational implications of Savage's theory of professions are considerable. The most germane implications for our purposes seem to be the following.

- The theory allows to narrowly define the area of operation of a profession because of its emphasis on core competencies — for example, pharmaceuticals, software, semiconductors, etc. — around which other capabilities and routines evolve and revolve.
- It allows to distinguish professions from other forms of organization, such as firms, because integration of ownership is *not* a *condicio sine qua non*.
- Professionals are autonomous and authoritative in their fields for their competencies allow them, on the one hand, 'to solve routine problems easily and non-routine problems routinely' (Savage 1994: 140) and, on the other, enable them to evaluate, and only be challenged by, other professionals. More concretely, they are independent yet interact in a coordinated and fertile fashion.
- Professions are decentralized networks in that there's not a central authority in command.[17] The 'organization' of a profession is guaranteed by the exchange of knowledge that reduces uncertainty and stimulates trust amongst members. Professions are thus self-organizing.
- Relatedly, there's the role played by reputation as a signalling of quality, *viz.*, reputation is a positive externality. Thus, professions can be interpreted as self-regulating organizations (a point we shall return to below).

The organizational workings of professions seem to well approximate, I think, some of the characteristics of the bazaar-style market for ideas that Raymond (2001) depicts in his descriptive analysis of open source. Similarly to open source, in fact, a profession is a capital network investing in network capital. Interestingly, the workings also seem to accord with the observation by Lerner and Tirole (2000: 8-9) that the core of the entire open source community seems to lie in the sophisticated individuals which compose it.

Yet, the theory of professions allows us to mostly illuminate one facet of open source: the supply-side. In order to offer a more complete story of open source, we need to extend the theoretical framework of professions to incorporate more explicitly demand-side considerations. We also need, moreover, to endogenize technology. To this end, it seems necessary to bring together the theory of professions with that of clubs, and to consider the role played by technology.

---

[17] See especially Langlois and Robertson (1995: ch. 5).

THE ADDITIONAL DYNAMICS OF A CLUB

In a seminal article published in 1965, 'An economic theory of clubs', Buchanan described and formalized the institutional properties of a new category of good (or product) lying between the public and private polar extremes, conventionally called shared good. The good is usually enjoyed only by members participating in a voluntary association — i.e., a club — whose membership may be regulated by some dues. The theory of clubs, in a nutshell, studies the different institutional arrangements governing the supply and demand of the shared good.

Since then, the vast literature on clubs has mostly devoted itself to the study of positive and normative issues at the macro level — for example, decentralization of government and fiscal federalism. But there have also been a few studies concerned with the firm. In particular, Antonelli and Foray (1992) propose a theory of technological cooperation among firms called 'technological clubs'. By means of a simple comparative static model, they suggest that firms will cooperate in technological endeavours only if the benefits of cooperation outweigh the costs. This is the traditional result we would expect under familiar club models, where the amount of shared good decreases as the number of users increases (cf. Buchanan 1965: 2-6).

But, interestingly, Antonelli and Foray also underline that this logic is reversed in the case of *network products*, namely, when 'the performance of the product as well as its utility increases with the increase of the community of users' (Antonelli and Foray 1992: 40). If there are for example possible network effects generated by the output, by the process of production or by the technology of production (or all of these), familiar exclusion/congestion effects caused by increased club membership may not hold.

Because of the necessary *standardization* that a network product requires, the possible exclusion/congestion effects generated by increased membership may be overcome by 'the overall growth of the aggregate demand for the production induced by network [effects]'. Therefore, 'the trade-off of the technological cooperation is reversed and now [a] firm [may choose whether] to enter a [technological club] and to standardize its own products according to the ratio between marginal costs of standardization and the marginal revenues of standardization' (Antonelli and Foray 1992: 43).

This begs the question of what the shared good is in our case. In the open source world, the shared good seems to be more than one: the *software* as well as the *capabilities* of production and of consumption. In light of our discussion so far, this claim should not be too surprising because, first, software *per se* is an ever-evolving capital structure that embodies knowledge (Baetjer 1998) and, second, because in the open source community both software and capabilities are nonrivalrous (Raymond 2001).[18] Indeed, if compared to proprietary software, open source would seem to assure an even more productive capital structure because of the free availability of the source code. *Knowledge and software are then not only the common (spontaneous) standards, but also the network products.*[19]

Now, were we in the presence of a more traditional organizational structure — such as one with a non-network product — we would have a congestion problem arising from the difficulty of capability transfer (cf. Ellig 2001). But because, as we noted, for open source esoteric knowledge is in reality common, the congestion is actually determined by the technological state of the art. (We shall return to technology presently.)

REPUTATION AND SHARED CONTEXT

These observations lead to another interesting issue. Open sourcers, we saw, are not in the trade to maximize profits. Although their first motivation to modify a program may originate from sheer need,[20] their utility functions for sharing, as Raymond repeatedly emphasizes, exhibit maximization of reputation; that is, attempting to deliver an ever better product maximizes reputation.[21]

Algebraically, we can illustrate the process in terms of quality improvements as follows:

$$q_t(S) = f(q_{t-1}(S), K_t, H_t, C_t),$$

where $q(S)$ is the quality of software S, K is knowledge, H is the complementary hardware, C is complementary software, and t is a time index. The utility (U) function of the open sourcers is:

$$U_t(S_q) = f(q_t, N_t(S_q)),$$

where N is the number of users of software S, q is quality, and t is a time index.

The qualitative property that this trivial illustration is trying to convey is the following. The endogeneity of reputation captured by the quality of software increases the user base (positive externality) and the 'utility' of the open sourcers on both sides of the demand and supply equivalence.

---

18 The suggested idea of sharing of capabilities shares some of the properties of user-based innovation and user-to-user assistance described in Lakhani and von Hippel (2000).

19 On open standards and networks cf. for example Garud and Kumaraswamy (1993) and West and Dedrick (2001).

20 Indeed, this is the first lesson offered by Raymond. 'Every good work of software starts by scratching a developer's personal itch' (Raymond 2001: 23, emphasis removed).

21 For example, the '"utility function" Linux hackers are maximizing is not classically economic, but is the intangible reward of their own ego satisfaction and reputation among other hackers' (Raymond 2001: 53).

This implies that many traditional organizational stories centred on, e.g., incentive alignment, monitoring, opportunism, and ownership structures are, at best, incomplete for they neglect true Marshallian external economies (or, if you prefer, knowledge spillovers) that act as, e.g., self-regulatory monitoring and coordinating systems. The 'poor beleaguered conventional manager is not going to get any [succour] from the *monitoring* issue; the strongest argument the open source community has is that decentralized peer review trumps all the conventional methods for trying to ensure that details don't get slipped' (Raymond 2001: 59, original emphasis).

Interestingly, the discussion brought us back to Hayek and to the problem of knowledge and its dispersion (Hayek 1948: chs 2 and 4; Jensen and Meckling 1998[1992]). That is to say, sometimes shared context may count more than hierarchy for the 'efficient' organization of production and exchange (Ghoshal *et al.* 1995).

ON THE ROLE OF TECHNOLOGY

The reader will have noticed by now that I have not yet said much about information and communication technology. We have talked about it in a standard comparative static fashion. But I have been deliberately vague about its endogenous role. This topic shall be briefly considered here.

Arguably, the role played in our story by information and communication technology is one of 'technological convergence' (Rosenberg 1963). Or, to update Rosenberg's notion somewhat, hardware and software represent a 'general purpose technology' (GPT) (e.g., Bresnahan and Traijtenberg 1995). Put simply, a GPT usually emerges to solve very narrow problems. Yet, in time its purposes diffuse to many other areas of application. For example, we have passed from the MIT PDP-1 minicomputer in 1961 to the Defence Department's Advanced Research Projects (DARPA) that created ARPANET, the first computer network, to today's personal computer and the Internet.

It would seem that even though GPT has greatly facilitated collaboration among a great variety and number of individuals, some of the economic interactions are very similar to what other forms of organization, such as professions and clubs, already do. But thanks to online interactions in real time, it would also seem that GPTs might ultimately give a comparative advantage to a profession/club mode of organization over one of hierarchy. Indeed, a fundamental reason for why classical often markets don't work well resides in the need to share rich information in real time (compare, e.g., Kogut and Zander 1992; Langlois and Robertson 1995).[22] In the specific case of open

---

22 As Raymond (2001: 224, note 10) observes in a related context, the 'open source community, organization form and function match on many levels. The network is everything and everywhere: not just the Internet, but the people doing the work form a

source it appears that the transition to cyberspace to share rich information in real time was so, as it were, smooth because there already existed a more-or-less well-defined core competence and culture.

POLICY CONSIDERATIONS

The first point to underline is probably that open source spontaneously solves the two fundamental organizational problems defined by Jensen and Meckling (1998[1992]: 103): 'the rights assignment problem (determining who should exercise a decision right), and the control or agency problem (how to ensure that self-interested decision agents exercise their rights in a way that contributes to the organizational objective)'. When specialized knowledge is symmetric, we saw, it spontaneously solves the agency problem by means of external economies (Savage 1994, 2000). And, just as Coase (1960) taught us, the ultimate result of this spontaneous interaction is a 'collocation' of production (and consumption) knowledge and decision power in the hands of those who most value it (Jensen and Meckling 1998[1992]; Savage 2000).[23] If this is so, then open source is not only a self-organizing organization, but also a self-regulating one where, as Savage (2000: 19) points out, 'self-regulation … means coordination of economic activity through voluntary association in an interdependent network, without interference from the government, and without resort to hierarchy'.[24]

Whenever organizational forms present rapid change because of their strong ties to technology, public policy issues are always thornier than usual. Indeed, historically, it seems that every time that there's the development of a new technology or production process, the government has to intervene in some fashion to regulate it or to extract rents from it. This point is well-encapsulated in the well-known catch-phrase attributed to Faraday. After Faraday was asked by a politician the purpose of his recently discovered principle of magnetic induction in 1831, he replied: 'Sir, I do not know what it is good for. However, of one thing I am quite certain, some day you will tax it'.

Since open source successfully developed in an environment of little government presence and it generated benefits well beyond its organizational boundaries,[25] the implications for policy are quite clear. The government must

distributed, loosely coupled, peer-to-peer network which provides multiple redundancy and degrades very gracefully. In both networks, each node is important only to the extent that other nodes want to cooperate with it'.

23 Miller and Drexler (1988), in a classic essay that greatly influenced Raymond (2001: 225-6), make a similar point. Compare also Baetjer (1998).

24 Cf. Raymond (2001: 57ff.).

25 The 'greatest economic contribution' of open source technologies 'may be the new services that they enable' (O'Reilly 1999: 34). The 'Red Hat Wealth Monitor' keeps track of the profits made by Red Hat thanks to open source software. This is an effort undertaken to encourage reinvestment in the community in order to try to generate even

be sensitive to economic activity that is spontaneously productive, and one way to guarantee this is to preserve spheres of autonomy.[26] Indeed, any intervention may suffocate the very motivation that drives these types of organization. At the same time, however, this is not to say that we should not think about the possibility of defining some Hayekian abstract rules for the interaction *among* new organizations as well as among new and more traditional types of organization (see Savage 2000).[27]

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

In a somewhat desultory fashion, I have attempted to describe some general organizational characteristics of bazaar-style software development. During the description a substantive lesson emerged. It appears that the emergence of spontaneous organization of work is facilitated mostly in those cases where the constituents at least to some degree already share productive knowledge or, if you like, where knowledge is already a standard. In the case of open source this is *a fortiori* so in that not only is the input (the esoteric knowledge) a sufficient statistic because of common absorptive capacity, but also because the output (the software, a network product) is itself essentially a standard-setting knowledge structure whose use is vast and whose reorganization is infinite. Relatedly, this suggests that in our story new information and communication technology — if now virtually indispensable — mostly performed the role of propagator rather than that of originator.

Without wanting to make too much of the point, we should also notice that thanks to new technology the organizational economics of open source now seem to be closer to the putting-out system.[28] The critical organizational difference between the putting-out and the bazaar being that in the latter there's no external authority controlling production.[29] That is to say that the

---

more wealth. Visit: <http://prosthetic-monkey.com/RHWM/> (accessed 21 March 2000). See also *Release 1.0* (1998); and cf. Lavoie ch. ?, this volume.

26 For instance, '[w]eb computing fundamentally depends upon open access because more contacts lead exponentially to more potential value creation. For example, Bob Metcalfe, inventor of Ethernet technology, asserts [that] the value of any number of interconnections — computers, phones, or even cars — potentially equals the square of the number of connections made' (D. H. Brown Associates, Inc. 2000: 8). This generalization should be readily contrasted to 'Brooks's Law' (Brooks 1975). As Raymond acknowledges, in '*The Mythical Man-Month*, Fred Brooks observed that programmer time is not fungible; adding developers to a late software project makes it later. He argued that the complexity and communication costs of a project rise with the square of the number of developers, while work only rises linearly. This claim has since become known as 'Brooks's Law' and is widely regarded as a truism. But if Brooks's Law were the whole picture, Linux would be impossible' (Raymond 2001: 49-50; see also 220-1, note 4). Compare Langlois (2001).

27 In many ways, this is the classic problem of increasing complexity as division of labor increases; but in a new guise. See especially Leijonhufvud (1989) and Baetjer (1998).

28 Cf. Leijonhufvud (1986) and Langlois (1999).

29 Or, to put it more precisely, authority is still present; yet it is internal in the sense that it is among peers, i.e., its legitimacy is mostly achieved by reputation.

organization of work of open source is one where the labour force is dispersed but connected by means of new technology, and whose product supervision is (spontaneously) assured by reputation effects.[30]

The economics of open source is very complex. My analysis has scratched the surface. Future studies should more closely scrutinize the relation between the structure of software, namely, its modularity, vis-à-vis the organizational structure delineated in the previous pages.[31] Further, they could explore the impact of type of licensing agreement on organization form, and study the relationship between the legal quandaries linked to traditional software[32] and possible implications for open source and its organization. And all these avenues of investigation would naturally lead into the very interesting issue of origin and evolution of organizational form[33] as well as into the one about the trade-off between coherence and flexibility of organization.

REFERENCES

Antonelli, C. and Foray, D. (1992) 'The economics of technological clubs', *Economics of Innovation and New Technology*, 2: 37-47.

Baetjer, H. Jr. (1998) *Software as Capital: An Economic Perspective on Software Engineering*, Los Alamitos, CA: IEEE Computer Society.

Bresnahan, T. F. and Trajtenberg, M. (1995) 'General purpose technologies: "engines of growth"?', *Journal of Econometrics*, 65 (1): 83-108.

Brooks, F. P. Jr. (1975) *The Mythical Man-Month: Essays on Software Engineering,* Reading, MA: Addison-Wesley.

Buchanan, J. M. (1965) 'An economic theory of clubs', *Economica*, N.S., 32 (125): 1-14.

Coase, R. H. (1937) 'The nature of the firm', *Economica*, N.S., 4 (16): 386-405.

---

30 The factory impression should be contrasted to more traditional comparative-static stories assuming exogenous inputs and outputs — hence constant technology and no knowledge combinatorics or growth (e.g., Malone *et al.* 1987; Gurbaxani and Whang 1991; Picot *et al.* 1996). These approaches are mostly influenced by the work of Williamson (e.g. 1985) for both method (i.e., the trichotomy among market, hybrid and hierarchy) and core variables (the transaction, asset specificity, contractual safeguard and opportunism); but cf. also Williamson (1996).

31 Kogut and Turcanu (1999) underscore the importance of modularity. For theories of organizational and technological modularity see, for example, Garud and Kumaraswamy (1993), Langlois and Robertson (1995: ch. 5) and Langlois (2001 and forthcoming).

32 On which compare Samuelson *et al.* (1994).

33 An issue embedded in all our discussion on which the Mozilla project by Netscape Communication, Inc. — that released the code of Netscape Communicator on March 31, 1998 (the first proprietary code to become open) — would be an interesting case study. See for example Rosenberg (2000: 33-8) and Raymond (2001: 61-3; 169-91). Also see: <http://home.netscape.com/browsers/future/whitepaper.html> (accessed 31 May 2001).

Coase, R. H. (1960) 'The problem of social cost', *Journal of Law and Economics* 3 (October): 1-44.

Cohen, W. M. and Levinthal, D. A. (1990) 'Absorptive capacity: a new perspective on learning and innovation', *Administrative Science Quarterly* 35: 128-52.

Cox, M. W. and Alm, R. (1998) *The Right Stuff: America's Move to Mass Customization. Federal Reserve Bank of Dallas Annual Report*. Online. Available HTTP: <http://www.dallasfed.org/htm/pubs/annual/arpt98.html> (accessed 14 April 2001).

cWare (nd) *The Linux Storm*. Online. Available HTTP: <http://www.cwareco.com/linux_storm.html> (accessed 19 March 2000).

D. H. Brown Associates, Inc. (2000) *Technology Trends Monthly*, May Newsletter. Online. Available HTTP: <http://www.dhbrown.com> (accessed 8 July 2000).

D. H. Brown Associates, Inc. (2001) *Technology Trends Monthly*, June Newsletter. Online. Available HTTP: <http://www.dhbrown.com> (accessed 9 August 2001).

de Solla Price, D. J. (1965) 'Is technology historically independent of science? A study in statistical historiography', *Technology and Culture* 6 (4): 553-67.

DiBona, Ch., Ockman, S. and Stone, M. (eds) (1999) *Open Sources: Voices from the Open Source Revolution*, Sebastopol, CA: O'Reilly & Associates, Inc. Also online. Available HTTP: <http://www.oreilly.com/catalog/opensources/book/toc.html> (accessed 21 July 2000).

Dolan, K. A. and Meredith, R. (2001) 'Ghost cars, ghost brands', *Forbes* (April 30). Online. Available HTTP: <http://forbes.com/global/2001/0430/068.html;$sessionid$FAXEMUIAABCTRQFIAGWCFFA> (accessed 7 May 2001).

Ellig, J. (2001) 'Internal markets and the theory of the firm', *Managerial and Decision Economics*, 22 (4-5): 227-37.

Eunice, J. (1998) *Beyond the Cathedral, Beyond the Bazaar* (May 11). Online. Available HTTP: <http://www.illuminata.com/public/content/cathedral/intro.htm> (accessed 19 march 2000).

Garud, R. and Kumaraswamy, A. (1993) 'Changing competitive dynamics in network industries: an exploration of Sun Microsystems' open systems strategies', *Strategic Management Journal* 14: 351-69.

Garzarelli, G. (2001) ‘Are firms market equilibria in Hayek’s sense?’, Centro di Metodologia delle Scienze Sociali Working Paper No. 74, Rome, Italy: Luiss-Guido Carli.

Ghoshal, S., Moran, P. and Almeida-Costa, L. (1995) ‘The essence of the megacorporation: shared context, not structural hierarchy’, *Journal of Institutional and Theoretical Economics* 151 (4): 748-59.

Gurbaxani, V. and Whang, S. (1991) ‘The impact of information systems on organizations and markets’, *Communications of the ACM* 34 (1): 59-73.

Hansmann, H. (1996) *The Ownership of Enterprise*, Cambridge, MA: The Belknap Press of Harvard University Press.

Hayek, F. A. von (1948) *Individualism and Economic Order*, Chicago: University of Chicago Press.

Jensen, M. C. and Meckling, W. H. (1998) ‘Specific and general knowledge and organizational structure’, in M. C. Jensen (ed.), *Foundations of Organizational Strategy*, Cambridge, MA: Harvard University Press. Originally published in Lard, W. and Wijkander, H. (eds) (1992) *Contract Economics*, Oxford: Blackwell.

Kogut, B. and Zander, U. (1992) ‘Knowledge of the firm, combinative capabilities, and the replication of technology’, *Organization Science* 3 (3): 383-97.

Kogut, B. and Turcanu, A. (1999) *The Emergence of E-innovation: Insights from Open Source Software Development*, Wharton School paper, University of Pennsylvania (November 15). Online. Available HTTP: <http://jonescenter.wharton.upenn.edu/events/software.pdf>. (accessed 25 February 2001).

Lakhani, K., and von Hippel, E. (2000) ‘How open source software works: “free” user-to-user assistance’, MIT Sloan School of Management Working Paper #4117 (May).

Langlois, R. N. (1999) ‘The coevolution of technology and organization in the transition to the factory system’, in P. L. Robertson (ed.) *Authority and Control in Modern Industry: Theoretical and Empirical Perspectives,* London: Routledge.

Langlois, R. N. (2001) *The Vanishing Hand: The Changing Dynamics of Industrial Capitalism* (August 7), Centre for Institutions, Organizations, and Markets Working Paper 2001-01, University of Connecticut, Storrs. Online. Available HTTP: <http://www.sp.uconn.edu/~langlois/Vanishing.html> (accessed 12 Augut 2001).

Langlois, R. N. (forthcoming) ‘Modularity in technology and organization’, *Journal of Economic Behavior and Organization*.

Langlois, R. N. and Robertson, P. L. (1995) *Firms, Markets, and Economic Change: A Dynamic Theory of Business Institutions*, London: Routledge.

Leijonhufvud, A. (1986) ‘Capitalism and the factory system’, in R. N. Langlois (ed.) *Economics as a Process: Essays in the New Institutional Economics*, New York: Cambridge University Press.

Leijonhufvud, A. (1989) ‘Information costs and the division of labour’, *International Social Science Journal* 120 (May): 165-76.

Lerner, J. and Tirole, J. (2000) ‘The simple economics of open source’, NBER Working Paper 7600 (March), Cambridge, MA: NBER.

Malmgren, H. B. (1961) ‘Information, expectations and the theory of the firm’, *Quarterly Journal of Economics* 75 (3): 399-421.

Malone, Th. W., Yates, J. and Benjamin, R. I. (1987) ‘Electronic markets and electronic hierarchies’, *Communications of the ACM* 30 (6): 484-97.

Miller, M. S. and Drexler, K. E. (1988) ‘Markets and computation: agoric open systems’, in B. A. Huberman (ed.) *The Ecology of Computation*, Amsterdam: North-Holland. Also online. Available HTTP: <http://www.agorics.com/agorpapers.html> (accessed 5 June 2001).

Mises, L. von (1981) *Epistemological Problems of Economics*, New York and London: New York University Press. Originally published 1933.

Nelson, R. R. and Winter, S. G. (1982) *An Evolutionary Theory of Economic Change*, Cambridge, MA: The Belknap Press of Harvard University Press.

O’Reilly, T. (1999) ‘Lessons from open source software development’, *Communications of the ACM* 41 (4): 33-7.

Parnas, D. L. (1972) ‘On the criteria to be used in decomposing systems into modules’, *Communications of the ACM* 15 (12): 1053-8.

Parnas, D. L., Clemens, P. C. and Weiss, D. M. (1985) ‘The modular structure of complex systems’. *IEEE Transactions on Software Engineering* 11 (3): 259-66.

Penrose, E. T. (1995) *The Theory of the Growth of the Firm*, 3rd edn, with a new ‘Foreword’ by the author. Oxford: Oxford University Press. Originally published in 1959.

Picot, A., Ripperger, T. and Wolff, B. (1996) ‘The fading boundaries of the firm: the role of information and communication technology’, *Journal of Institutional and Theoretical Economics* 152 (1): 65-79.

Polanyi, M. (1966) *The Tacit Dimension*, London: Routledge.

Raymond, E. S. (2001) *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*, revised edn, Sebastopol, CA: O’Reilly & Associates, Inc. Also online. Available HTTP: <http://www.tuxedo.org/~esr/> (accessed 5 February 2000).

*Release 1.0, The Open-source Revolution (*1998, 19 November). Online. Dead HTTP: <http://www.edventure.com/release1/1198.html> (accessed February 10, 2000); now available in PDF format from HTTP:

<http://release1.edventure.com/sampleissues.cfm> (accessed 19 April 2002).

Richardson, G. B. (1998) *The Economics of Imperfect Knowledge: Collected Papers of G. B. Richardson*, Cheltenham, UK: Edward Elgar Publishing, Inc.

Rosenberg, N. (1963) 'Technological change in the machine tool industry, 1840-1910', *Journal of Economic History* 23 (4): 414-43.

Rosenberg, D. K. (2000) *Open Source: The Unauthorized White Papers*, Foster City, CA: M & T Books.

Ryle, G. (1971) 'Knowing how and knowing that', In *Idem (*ed.) *Collected Papers*. *Volume II: Collected Essays, 1929-1968*, New York: Barnes & Noble, Inc. Originally published in *Proceedings of the Aristotelian Society* (1946) 46.

Samuelson, P., Davis, R., Kapor, M. D. and Reichamn, J. H. (1994) 'A manifesto concerning the legal protection of computer programs', *Columbia Law Review* 94 (December): 2308-431.

Savage, D. A. (1994) 'The Professions in theory and history: the case of pharmacy', *Business and Economic History* 23 (2): 129-60.

Savage, D. A. (2000) 'Self-regulation and the professions', unpublished manuscript (November).

West, J. and Dedrick, J. (2001) 'Proprietary vs. open standards in the network era: an examination of the Linux phenomenon', *Proceedings of the Hawai'i International Conference on System Sciences* (HICSS-34) (January 3-6), Maui, Hawai'i, IEEE. Online. Available HTTP: <http://www.gsm.uci.edu/~joelwest/Papers/WestDedrick2001.pdf> (accessed 5 June 2001).

Williamson, O. E. (1985) *The Economic Institutions of Capitalism*, New York: The Free Press.

Williamson, O. E. (1996) 'The fading boundaries of the firm: comment'. *Journal of Institutional and Theoretical Economics* 152 (1): 85-7.