Taming of ‘Openness’ in Open Source Software Innovation

Mehmet Gençer[1], Beyza Oba[2]
mgencer@cs.bilgi.edu.tr, boba@bilgi.edu.tr
[1]Istanbul Bilgi University, Department of Computer Science
[2]Istanbul Bilgi University, Department of Business Administration

Abstract

Open innovation, and particularly open source software is associateded with openness in terms of participation to the innovation process. Despite the prominent image of openness to participation, however, a variety of measures are taken in large scale innovation ecosystems to tame the potentially chaotic activity and growth in bottom-up innovation, and align contributions with the strategic priorities of major stakeholders. Such taming rests on the dual desires to unleash the proven power of bottom-up innovation on the one hand, and to ride it in certain directions on the other. It emerges in the form of a variety of governance mechanisms, licensing strategies, organizational practices, etc., but exhibit certain common themes. In this paper, by drawing on several examples of large scale open source software ecosystems, we demonstrate that methods employed for taming exhibit a considerable isomorphism. By drawing also on examples of failures, we further suggest that the computing industry is settling on what we call Pax Innova. It is a coherent set of practices which retain virtues of bottom-up innovation while allowing major stakeholders investing into the ecosystem to selectively support activities, and introduce top-down corporate discipline such as quality assurance measures. The choice of available practices depend on initial conditions (i.e. whether the ecosystem is led or only supported by private firms) but nevertheless shaped around the same set of requirements.

1 Introduction: the case for taming openness

Collaborative innovation is becoming increasingly commonplace in many industries, ranging from biotechnology to computer hardware/software. Even the most powerful business organizations are struggling to respond to quick shifts in consumer demands, in a creative and agile manner. The consequences, it appears, frequently involves seeking innovation beyond its traditional locus of internal R&D labs. Various sizes of business organizations pursue bilateral or multilateral arrangements to actively cultivate inter-organizational networks of collaborative innovation.

It has been more than two decades since this phenomenon has found its way to organizational research, and collaborative business strategies are claimed a place next to competitive ones (Jorde and Teece, 1989; Ring and van de Ven, 1994); but the phenomenon seems to be unfolding and evolving with an increasing speed, presenting as many questions as answers for organizational research. We are yet to proceed from retrospective accounts to instrumental answers for striking the balance between competition and cooperation.

Among several other industries, software industry has been one in which business perception towards collaborative innovation strategies is changing rapidly. It appears to be the case that adaptation is replacing planning as the fundamental principle of innovation in software technology development. In the word of Google CEO Eric Schmidt "The only way to ensure it [innovation] can flourish is to create the best possible environment-and then get out of the way. It's a question of learning to live with a mess"[1]. Within this changing business mindset, Open Source Software (OSS), once a marginal movement, is finding its way into mainstream inter-organizational innovation practices in the software industry. OSS provides proven methodologies which stimulate software product evolution in expense of the predictability of process (opposite of the established practices of software engineering), and it is known to promote product interoperability which in turn eases the task of coordination in networks (Behlendorf, 1999; Benkler, 2005). Leading players in the computing industry, such as IBM, Apple, and Google, actively cultivate open source innovation networks and/or use software technologies coming out of these networks as components in their key products or operations (West, 2003). Even notable exceptions such as Microsoft is experimenting with the open source model.

Despite the rhetoric of openness surrounding these business practices, however, considerable effort is put into taming the 'mess'. Within the multitude of often conflicting business agendas, each participating business organization in a collaborative innovation system faces the issue of turning the collaborative outcomes into competitive advantage for itself. On the one hand, the basic tenet of combining knowledge resources through the networks is to overcome rigidities of innovating in isolation (Leonard-Barton, 1992), to maintain competitive edge with possibly radical innovations that may come out of the improved vividness in a collaborative network, rather than merely incremental ones that are the likely outcomes of isolated innovation (von Hippel, 2006). On the other hand, a business organization would like to drive this emergent, rather than planned, innovation process in a direction that makes sense for its own priorities. While openness in OSS is associated with unleashing the bottom-up innovation, blending it into a business strategy necessarily faces the dilemma of re-leashing this innovative power to make it business-wise useful.

In this exploratory study we investigate several cases of collaborative innovation projects based on OSS, and attempt to identify common patterns in organisational structures and processes which address the duality of empowering bottom-up innovation while at the same time imposing requirements on its direction and outcomes in accordance with business priorities of corporate participants. Our focus is to lay out the elements of an emerging practice and organisational form, how these elements address the problems of collaborative innovation in software industry, and consequently, how one may judge a particular combination of those on the bases of both institutional acceptance and suitability to the requirements of inter-organisational arrangement. In the following sections we first present the conceptual background on inter-organisational innovation in general, and OSS in particular. We then present several cases of well known, large-scale OSS-based collaborative innovation projects from the software industry, and identify common mechanisms and practises used for taming of openness. Furthermore, based on cases of business failure, we elaborate on the arguments of other researchers who claimed that emergence of OSS signifies the end of proprietary software innovation model which has prevailed the industry so far.

2 Theoretical and Historical Background

2.1 Inter-organisational collaboration

In addressing how innovations are made and how they create competitive advantage, Teece (1986; 1989) questioned the role of corporate R&D facility as the driver of innovation. By placing design, rather than science, at the center of innovation process, Teece suggested that richness of collaborative linkages with other firms, universities, etc., is the major driver of innovation capability. Corporate R&D work within firm boundaries complements this capability, often by introducing modest improvements on innovations which appeared elsewhere. Furthermore, network linkages and positioning were important in terms of complementary assets which is an important element of how an innovation creates competitive advantage in the presence of imitators.

Various streams of business and economics research has approached the issue of collaborative innovation since then. Inter-firm networks are ascribed roles ranging from being a prerequisite for commercialising innovations (Kogut, 2000) to itself being the locus of innovation (Powell et al., 1996). Within this networked view of the business firm various aspects of economic performance were subjected to empirical research: structure of firm network ties in relation to its innovation outcome (Ahuja, 2000), inter-organisational learning (Powell et al., 1996; Lane and Lubatkin, 1998; Nooteboom, 2008), role of network ties for firm survival (Oliver, 1994), etc. In addition, some research addressed the matter with an extra-organisational perspective, considering a range of problems such as the content and structure of inter-organisational networks of innovation and production in technological systems or districts (Langlois, 1990; Castilla et al., 2000; Antonelli, 2000), and adverse effects of protective regimes for realising collaborative innovation in such systems (Heller and Eisenberg, 1998; Pisano, 2006; West and Gallagher, 2006).

While collaborative innovation grows both as a phenomenon and a research stream, its problems also become evident. Initiation of inter-firm relations relies on a variety of antecedent factors (Oba and Semercioz, 2005), and development of relations and mutual trust that sustain them is hard to predict or control (Ring and van de Ven, 1994). Developing relations entail sharing critical knowledge (Gächter et al., 2010) and taking risks (such as committing to relation specific investments). On the other hand, as emphasised by open innovation framework (Chesbrough, 2006; Chesbrough and Appleyard, 2007), failure to establish collaborative innovation relations often results in under-utilisation of innovations created inside the firm. Traditional business strategy of constructing barriers to prevent knowledge leaks also prevents knowledge intake and impedes combination of capabilities.

The contemporary business environment thus leaves the business organisation in a dilemma: unit of competition is still the business organisation but inter-organisational collaboration is an important element of competition. Despite the risks, collaborations happen and actively sought for. The traditional form of inter-organisational collaboration has been bilateral arrangements between two firms. Such arrangements involve some combination of product/technology licensing, co-development of technology, etc. Recently, however, a multilateral form of inter-organisational collaboration is becoming commonplace. The multilateral form appears with a variety of names including the more traditional ‘consortia’, ‘alliance’, ‘platform’, or more recent ‘ecosystem’. Some examples are Human Genome Project, Globus Alliance, and Android Platform. Such alliances became so important in competition that the CEO of a mobile devices producer would describe the situation as : “the battle of devices has now become a war of ecosystems”[2]. Therefore, in today’s market conditions, many business organisations have a wealth of motives to enter multilateral systems of inter-organisational innovation.

On the other hand multilateral systems lack the established mechanisms of bilateral systems. Bilateral collaborative arrangements have a longer history with a repertoire of mechanisms such as licencing agreements or joint ventures. Thus, the business environment is at an infancy phase in terms of institutionalisation of collaborative innovation networks. Designing such inter-organisational systems by finding governance and coordination mechanisms, property regimes, etc., to make such networks function in a way which reduces risks for participants and increases predictability of outcomes remains an experimental innovation task in itself. However, while such an organisational design task is expected to produce something new for a new problem, it is at the same time must make sense, i.e. must be recognisable and understandable (Krippendorff, 1989) to participants of the inter-organisational system. In such a situation where things are uncertain but collective decisions are to be made, one must resort to proven solutions that make immediate sense for everyone. Thus organisational designs of collaborative innovation networks may have different starting points and evolutionary paths depending on the industry and its established practises, although they solve similar problems. Here we will only consider the case of software industry and open source software.

2.2 Open Source Software

Open Source Software is both widely cited in collaborative innovation literature, and at the same time treated specially. There are a variety of reasons for this special treatment. First, OSS has developed outside the business world, in communities spanning academia and young professionals (McKusick, 1999; Raymond, 1999). While it has recently found its way into business strategies in the software industry, it did come with a baggage of historically developed methodologies and property regimes of its own. Thus it exhibits a different history in its second life which involves experimentation of businesses to adapt it into a commercially viable form. Second, the software field generally lacks patents which is the norm of intellectual property protection in most industries. Therefore, unlike biotechnology, for example, once disclosed software source code can no longer be protected against appropriation. For this reason, OSS based collaborative innovation networks use commons based property regimes, rather than bilateral or multilateral licensing arrangements common in similar networks in biotechnology. The so called open source licenses essentially give the right to use software to everyone. The historical license type of OSS was the public license, which further mandated disclosure of any modifications to public licensed software (or even other software that is integrated with it as part of this modification). However, years of experimentation with OSS in the software industry resulted in widespread use of more liberal licenses which encourage collaboration while allowing usage in commercial products in the traditional sense. The model based on liberal licenses is commonly referred as open source software, where the one based on more restrictive public licenses is referred as free software. In both models the commons based property regimes combined with transparency of OSS software development methodology makes OSS an area where the openness in innovation is very high; both the process and outcome of innovation is open (Huizingh, 2010).

A considerable share of research on OSS focused on community led projects, which is the representative form of its pre-business history. These studies explored internal workings of OSS communities such as motivations of individuals for contributing to public software (Hertel, 2003; Lakhani and Wolf, 2005; Lakhani and von Hippel, 2003; von Hippel, 2005), and highlighted meritocratic leadership and community norms as the basis of coordination and control (Kogut and Metiu, 2001). Another stream of OSS research investigated recent business experimentation with the OSS model (West, 2003; Samuelson, 2006), business models underlying these experiments (Krishnamurthy, 2005), and implications of the rise of OSS model for software industry (von Hippel, 2001; Bonaccorsi et al., 2006; Economides and Katsamakas, 2006). More recently several researchers claimed a fundamental shift in the software industry as a result of OSS model's popularity, using various labels for the phenomenon such as 'OSS version 2' (Fitzgerald, 2006) and 'OSS generation 2' (Watson et al., 2008). Although limited in scope, some recent empirical findings (Hauge et al., 2008) seem to support Fitzgerald's (2006) claim that these developments signify the end of proprietary model and a paradigm shift in the software industry.

Despite the increasing research interest in OSS, the literature is limited in terms of understanding emerging forms and processes in OSS based collaborative innovation in relation to business strategies. Open innovation framework, for example, has an ontological basis which consists of the organizations, their relations and knowledge content of these relations, allowing one to approach competitive and collaborative issues simultaneously. However, it is often the case that organizing collaborative innovation requires creation of practices and even formal entities for governance of such activity, which are associated with the network itself rather than particular actors or relations. What is highlighted by the community oriented approach of some OSS research becomes invisible under the actor oriented approach of open innovation framework, and vice verse. This may well be the reason behind recent preference of the term 'ecosystem' over 'network' in business OSS jargon which is more conducive to speak about content and design of inter-firm collaborative innovation networks, while blending relatively well with open innovation thinking and competitive aspects. Many business organisations in the software industry seem to prefer the ecosystem concept in reference to collaborative innovation networks they lead or participate. Among other things the term signals ‘sustainability’ to potential adopters of the technology that is created within the collaborative innovation ecosystem.

OSS model provided the software industry with a design recipe that made sense to potential participants of the inter-organisational collaboration ecosystems to be established. Increasing numbers of such systems appearing since 2000s have mimicked the original model and one another’s variations, while introducing incremental changes to make it more suitable to business environment (see DiMaggio and Powell 1983 for mimetic behaviour and institutional isomorphism). What is copied or modified includes property regimes, governance mechanisms, coordination processes, and even mutual expectations of participants. The historical development from free software to open source software, then its combination with the ecosystem metaphor, signals the major turns in the developmental path: the first reflecting the concern with appropriability/commercialisation, and the latter reflecting the concern with productivity/sustainability. However, the differences among emerging stereotypes can be important in understanding the role of each organisational element for the working of collaborative innovation in a variety of contexts.

3 Research Design

Our goal here is an exploration of emerging inter-organisational innovation system designs in software industry based on OSS model and identification of common elements in these designs. There are various elements in these designs which we have included in our investigation because they are related to (i) inter-organisational sharing and coordination of responsibilities and outcomes, (ii) making common sense and appealing to the participants for collective decision making, or (iii) influencing of goals or outcomes by single participating firms.

An apparent one of these elements is the property regime, with its corresponding software *licensing*. Since license differences can introduce serious legal issues (de Laat, 2005), most projects use only one OSS license, or at best a few closely compatible ones. Another element is formal *governance mechanisms*, or their absence. Often projects create formal bureaucracies for governance as they grow, although the level of formalism varies between them, or even some severely object such mechanisms. Decision making in OSS projects is commonly understood in terms of either (i) *setting the major design* *goals* of software product by selecting among proposed features, or even creating subprojects (i.e. bundles of features) when necessary, or (ii) *promoting* some contributors to ‘committer’ status who has the right to change or accept changes to master copy of software source code. OSS projects are open about both processes, although they are rarely formal, and with varying degrees when they do so. Finally, how single *businesses’ influence* the innovation network is only partially accessible when it happens through governance mechanisms.

In selecting research cases we have chosen among only well-known and large scale OSS projects which are influential cases for the software industry. Furthermore we have chosen those for which relevant data is publicly available. We have included cases from two major categories of community-led and business-led OSS projects. Some OSS projects start their lives as small non-commercial projects led by a small community of developers, but later meet corporate interest. Some other projects are the other way around: they are initiated by one or a group of firms, but later may enjoy support of individuals from professional communities.

Complimentary to he above, we have also selected some major and recent cases of competitive disadvantage in relation to failure to embrace open source model. We have compiled brief, unstructured statements of these cases. Although limited in scope and extend, these cases are used to enrich our interpretation of the findings, and to check for position of the OSS model within the software industry.

4 Cases and findings

The OSS cases used in our investigation and summary of findings about each are as follows:

Apache Most widely used web server software which has a liberal license allowing use in commercial products. Apache is a community project since its start in 1999 but enjoyed a lot of corporate support early on from firms like IBM who use the technology in some of their products. Apache is governed by the Apache Group which later became a foundation. New activities are put through an incubation process for taming “providing guidance and support to help each new product engender their own collaborative community, educating new developers in the philosophy and guidelines for collaborative development as defined by the members of the Foundation”[3].

Linux A major operating system software whose development is led by a vast global community, with a public license (Genaral Public License, GPL) limiting commercialization. But it is used as infrastructure element by many firms successfully. Some key developers were later hired by leading companies in the industry. It lacks any formal governance body, and led by its originating leader Linus Torvalds since 1991, and has a closed and small leadership team. Yet the accelerating corporate contributions has prompted creation of a foundation in 2007 dedicated to fostering the technology, and hosts several events where influential community members are brought together.

Eclipse Very successful software development platform initiated by IBM in 2001. The alliance has later became a foundation which oversees the development ecosystem in a more transparent way. Foundation members are leading firms of the computing industry who pay dues and dedicate developers to the project. Contributing individuals are promoted to the committer status by existing committers, provided that they demonstrate 'discipline and good judgement'[4]. New projects are put through an incubation process, similar to Apache case. The liberal license of Eclipse software was later refined by the foundation and made compatible with GPL to facilitate reuse in appropriation, along with changes in code acceptance policies.

Mozilla A project with very successful web browser and e-mail software. It was initiated by the firm Netscape upon its failure to compete with Microsoft's Internet Explorer. It's license has gone through several changes from one that explicitly favored Netscape, to a more standard liberal license compatible with the broader open source ecosystem (de Laat, 2005). Mozilla considers itself as a "hybrid organization, combining non-profit and market strategies to ensure the Internet remains a shared public resource"[5]. Mozilla uses an incubation process similar to Apache for taming new projects.

GCC The GNU Compiler Collection is a fundamental piece of software which is used for compiling many other software for a variety of platforms, and is licensed with GPL. Its development, started in 1987, was confined within the GNU team led by Richard Stallman, a prominent figure of the open source movement. As it became a fundamental technology for many firms targeting the UNIX platform, a steering committee was formed in 1999 with representatives from leading firms and universities, but the project leadership emphasizes that the committee members represent communities, not their employers.

Android This recent open software platform for mobile devices had a huge impact in the market in only three years following its announcement in November 2007. Vari-

	Community led	***Corporate led***
Licensing	GPL, targets largest adoption	liberal, balances adoption and appropriation
Governance mechanisms	not formal, meritocratic	foundation/bureaucratic, transparent
Acceptance of new projects	not transparent, left to leadership	incubation process
Promotion of developers	meritocratic	meritocratic
Corporate influence	hiring lead developers, or through committees	through bureaucracy

Table 1: Mechanisms and practices of taming in corporate led and community led OSS projects

ous device producers switched to Android, further strengthening its acceptance and contributing features. The innovation ecosystem is led by Google, and formalised as the Open Handset Alliance with 81 member organisations from various segments of the mobile industry. The software is licensed using the liberal OSS license used in the Apache project. Although promotion policies are transparent, these are less meritocratic compared to more mature projects like Eclipse; for example it is only the project leaders (who happen to be Google employees) who promote others, etc. Therefore corporate influence is also through this hierarchy. It is unclear how new projects are approved.

Feature of these cases in term of licences used, governance mechanisms, creation of new projects, developer promotion, and corporate influence mechanisms are summarized in Table 1 separately for community and corporate led OSS projects. The difference in property regimes and corresponding licensing schemes reflect the difference of concerns with wider adoption (in community-led type) and suitability for use in commercial products (in corporate-led type). The experience seems to have shown that replacement of anti-intellectual property regime of free software movement with more liberal licenses is generally welcome. Despite minor variations liberal licenses (with archetypes such as Apache and Mozilla licenses) seems to be the established norm for new OSS projects.

In comparing the two types, it is interesting to note that community led projects are less transparent in terms of governance. This situation may look somewhat counter intuitive at first, but there are business strategic advantages of transparent governance in and inter-organisational setting. Such transparency serves well for directing community attention and setting the goals. Fitzgerald (2006) notes how such transparency is utilitarian for avoiding the strategic planning vacuum of community-led projects, in which software development followed ‘an itch worth scratching’. Corporate-led projects aim to be inclusive and flexible in their planning process, but nevertheless such planning vacuum is avoided by careful outlining of targeted software features and development road-maps, and making hem widely available. Hence using transparency as an instrument for directing community attention and effectively setting goals.

In a similar vein to governance mechanisms, corporate influence is more transparent and works through formal bureaucracy in corporate-led projects, as in the example of Eclipse where each major participating business dedicate certain amount of human resources (among financial resources) to project and have seats in decision-making bodies. On the other hand corporate influence in community-led projects works through hiring lead developers or having seats in governance committees. Original leadership in community-led projects rarely waive their command (at least explicitly or formally) in favour of new corporate stakeholders. Thus making the new players part of the old game where they rule, rather than subordinating themselves to a newly generated bureaucracy. Generally, as Watson (2008) notes it's rather the amount of contribution rather than formal ownership that is the basis of control and influence.

Again similarly, creation of new sub-projects is handled more transparently in corporate-led projects where the process is given names as 'incubation period/process'. In Apache and Eclipse examples, this period involves tempering of new project and its leadership to make processes and software products meet same quality standards, etc. In community-led projects, on the other hand, all is decided by meritocratic leadership. There may even be cases where leadership makes decisions despite strong opposition (Gençer et al., 2006), which is hard to imagine in the practises of corporate-led projects.

One common element among the two types is meritocratic promotion of developers. Meritocratic promotion is seen as an important basis of OSS model and seems to be fully adopted in corporate-led OSS projects. Besides the somewhat exceptional case of Android, both community-led and corporate-led projects adopt voting-based schemes for developer promotion. Contributors who demonstrate good skills and adapt to community norms are voted into ranks by existing committers. Norms (re)articulated by existing meritocracy thus becomes an important element of taming within collaboration domains. We must note that Eclipse project had less transparent practises in this (and other) respects at its beginning, and grew more transparent as the alliance has expanded and trust building between partners demanded such transparency. Therefore one can expect Android also to evolve into a more transparent project as it grows.

These findings provide a picture of what is being tamed in business adoption of OSS, and in what direction. Most clear OSS element that is left intact in taming is meritocratic promotion of developers. Watson (2008) notes how OSS enables private businesses to access high quality talent base, and helps reduce hiring risks. Thus retaining the meritocratic promotion practice not only keeps development quality high as experienced long before corporate-led OSS, but also provides access to talent that is relevant. In similar vein, adjustments to OSS licensing (see de Laat 2005 for e review) was kept at a point where it both attracts high quality and volume of contributions while allowing them to be used in commercial products. On the other hand formal governance and goal setting (new project acceptance) practises are specific to corporate-led era of OSS and appears to be still experimental. Some popular success stories like Eclipse are likely to set emerging norms in this respect. Furthermore, comparison of the Android case to others like Eclipse and Apache suggest that expanding the support base of collaborative innovation is associated with increasing transparency in governance structures and practises. Such transparency not only improves trust among partners, but also makes the process more predictable. In addition it allows participants to cast a direction to goals and outcomes of the collective process, thus contributing to the taming of OSS process.

Increasing use of formal governance bodies and practices in OSS-based collaborative innovation introduces an organisational innovation into the software industry. In a business environment where there are established bilateral forms such as joint ventures, such multilateral forms are rather new. This phenomenon further highlights the need for ontological commitment to network in inter-organisational collaboration research, since the networks start to become concrete entities with some sort of controlling bodies and norms of their own.

4.1 Cases of non-OSS

We also draw on some well known examples of failure to embrace open innovation strategy. Here our compilation of case summaries are not structured, and aims to capture brief historical background of declining competitive advantage in relation to OSS related developments in relevant market segments:

Sun The company's most prominent technology, Java, has been distributed freely to promote adoption. But although Sun used their so called 'community process' to incorporate industry demanded features into their Java technology, the company essentially maintained a hard handed approach for controlling it. While companies like IBM enjoyed considerable increase in the revenues from Java technology, Sun, despite their ownership of the Java brand, trademarks, and patents, has failed to turn this advantage to income. Although the company finally decided to adopt open source strategy in the face of this competition, this move came rather late and the company was finally sold to Oracle following its decline over the years.

Microsoft The company has made several moves towards embracing open source strategy starting at 2004. However, these moves appear mainly a PR activity rather than a serious strategy, as the company officials note that it does not involve any core product[6]. Google trends show that the excitement around Microsoft's open source move is fading out[7]. Despite its firm position in the PC software market, the company's position seems to be under threat. In May 2010 stock market value of Apple has surpassed Microsoft for the first time[8]. Although Apple does not have a big share in the PC market it has been successful in offering innovative products in the mobile segment, and adopted open innovation strategy for many years at the very core of its technologies(West, 2003). Similarly the open source Android technology by Google has been very successful in the mobile market, further limiting Microsoft's future prospects for expanding beyond the shrinking PC market. The shifts in relative positions of these players suggest the advantage of open source strategy in the new market landscape.

Nokia Having maintained a strong market leadership for so long, the company's market share and profits are on the fall with entry of Apple and Android into the mobile market. Nokia has a history of collaborating with industry partners, such as for developing the Symbian operating system used in smartphones. However, the current situation shows that the Android alliance is quickly sweeping out the Symbian from mobile devices. Nokia's recent reaction to these developments was to change the troubled mobile OS, by announcing partnership with Microsoft[9] (instead of, for example, being a late participant in Android alliance).

It is hard to substantiate an argument that OSS model invalidates the previously prevalent proprietary software model, based on isolated cases. However, they do indicate significance of this claim voiced by various researchers (Fitzgerald, 2006; Watson et al., 2008). All are cases in which use of OSS based collaborative innovation strategy created a vacuum which left ostensibly established players who fail to adapt in a competitive disadvantage. Furthermore, the case of Nokia suggests that collaborative innovation strategy based on non-OSS models fail to grow in the face of OSS-based alternatives, despite first mover advantages.

5 Discussion: Pax Innova

Some authors have already pointed to fundamental shifts in OSS processes, and noted that these indicate alteration of ground rules in both the OSS movement and the software industry (Fitzgerald, 2006). Our findings further this argument and present the organizational elements which are associated with the shifts in the open source process towards more corporate involvement. There appears to be an emergent pattern of proven mechanisms, strategies, and processes, the totality of which we refer here as Pax Innova to facilitate discussion.

Our aim in introducing a new term is to pay due attention to the state of the change in software sector. Using a term like OSS v2 indicates a fundamental change in the OSS, but diverts attention from the business strategic roots of the overall change it intends to refer to. We define Pax Innova as the new attractor of software industry, an emerging pattern of inter-organisational design of collaborative innovation which is in the process of institutionalising. We have emphasised how such a new design develops from existing designs, in incremental steps such that each modification is recognisable and makes sense to stakeholders. Yet, the emergent attractor is radically different from its dual origins (of OSS and proprietary models) and deserves a new name.

In the background of these developments is the increasing speed of innovation, driven by variety and volume of demand, high connectivity of devices, and high portability of information. This makes it hard for even the biggest players to control or supply complex products on their own. When a complex system such as the software industry is on the verge of fundamental transformation, various marginal practises can become common as they fit well into the new environment. Greater acknowledgement of the need for collaborative innovation is one part of Pax Innova, and OSS seems to be the organizational model of choice. Its emergence is probably triggered by the success of popular community products such as Linux and Apache, but its true admission to business strategy practises went through a series of experiments. And it went all well.

As far as this emergent combination of solutions work, it can be expected to be the first choice of collaborative innovation ecosystems which will be established in the software industry. Nokia's Symbian project might as well continue to be as innovative as Android, but the players of the industry seem to relate much more easily to OSS model at the moment. Phrasing the current situation rather dramatically, OSS makes a lot of sense, and make all others look like nonsense.

Elements of Pax Innova solution are different from those common in corporate hierarchies. Rather than targeting a top-down control or planning, they are instruments of taming. It is through these elements that an OSS ecosystem continues to be an emergent system which retain advantages of bottom-up innovation, while at the same time its major stake-holders can “invite it to emerge”(Jelinek, 2004) in certain ways rather than others.

Finally, a second meaning we can associate with Pax Innova is the fact that free software and open source software threads reaching a settlement in which their focus shifts to co-endowment rather than arguing over their differences. The community now enjoys more interest and more resources, while the corporate movement keeps finding ways to turn community experience to viable business practices.

6 Conclusion

In this study we have focused on emergence and role of practices and governance structures that are associated with taming of collaborative innovation networks based on OSS model, and attempt to identify their common features, and requirements calling upon their emergence.

We have used several cases of community led and corporate led OSS projects, and attempt to identify common patterns in organizational structures and processes which address the duality of empowering bottom-up innovation while at the same time imposing requirements on its direction and outcomes in accordance with business priorities of corporate participants. We necessarily take the network itself as the ontological basis of our investigation, but this choice is purely pragmatic, and the requirements of governance mechanisms and practices are investigated in relation to their contribution to quality of inflow and outflow of organizational knowledge in open innovation, which is in turn actor oriented. We contend that community led and corporate led OSS ecosystems differ, where (1) practices and formal arrangements for taming is more explicit and transparent in the latter, (2) meritocratic basis of developer promotion is retained in both, (3) means of asserting influence by corporations is different in both, but considered legitimate as far as the meritocratic basis is retained.

By also referring to cases of failure to embrace OSS innovation and its negative consequences, we further suggest that the computing industry is converging upon OSS innovation.

Notes

[1] Interview published at Information Week e-zine, http://www.informationweek.com/news/global-cio/showArticle.jhtml?articleID=227600039

[2] Wall Street Journal, Feb. 9th, 2011: http://blogs.wsj.com/tech-europe/2011/02/09/full-text-nokia-ceo-stephe

[3] http://incubator.apache.org/incubation/Incubation_Policy.html

[4] Eclipse committer policy

[5] http://www.mozilla.org/about/organizations.html

[6] http://www.seattlepi.com/business/179256_msftopen25.html

[7] http://www.google.com/trends?q=microsoft+open+source

[8] http://techcrunch.com/2010/05/26/apple-microsoft-market-cap-2/

[9] http://www.bbc.co.uk/news/business-13154794

References

Ahuja, G. (2000). Collaboration networks, structural holes, and innovation: A longitudinal study. *Administrative Science Quarterly 45*, 425–455.

Antonelli, C. (2000, August). Collective knowledge communication and innovation: The evidence of technological districts. *Regional Studies: The Journal of the Regional Studies Association*, 535–547.

Behlendorf, B. (1999). *Open Sources: Voices from the Open Source Revolution*, Chapter Open Source as a Business Strategy. O'Reilly.

Benkler, Y. (2005, May). Coase's penguin, or, linux and the nature of the firm. In R. A. Ghosh (Ed.), *CODE : Collaborative Ownership and the Digital Economy*, pp. 169–206. Cambridge: The MIT Press.

Bonaccorsi, A., S. Giannangeli, and C. Rossi (2006). Entry strategies under competing standards: Hybris business models in the open source software industry. *Management Science 52*(7), 1085–1098.

Castilla, E. J., H. Hwang, E. Granovetter, and M. Granovetter (2000). *The Silicon Valley Edge : a habitat for innovation and entrepreneurship*, Chapter Social Networks in Silicon Valley, pp. 218–424. Stanford University Press.

Chesbrough, H. (2006). *Open Innovation: Reaching a new paradigm*, Chapter Open Innovation: A new paradigm for understanding industrial innovation. Oxford University Press.

Chesbrough, H. W. and M. M. Appleyard (2007). Open innovation and strategy. *California Management Review 50*(1), 57–76.

de Laat, P. B. (2005, December). Copyright or copyleft?: An analysis of property regimes for software development. *Research Policy 34*(10), 1511–1532.

DiMaggio, P. J. and W. W. Powell (1983, May). The iron cage revisited: Institutional isomorphism and collective rationality in organizational fields. *American Sociological Review 48*, 147–160.

Economides, N. and E. Katsamakas (2006). Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science 52*(7), 1057–1071.

Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly 30*(3), p587 – 598.

Gächter, S., G. von Krogh, and S. Haefliger (2010, June). Initiating private-collective innovation: The fragility of knowledge sharing. *Research Policy*.

Gençer, M., B. Oba, B. Özel, and V. S. Tunalıoğlu (2006). Forking: The gpl coherent technology for flexible organizing in foss development. European Group of Organizational Studies 2006 Colloqium in Bergen, Norway.

Hauge, O., C.-F. Sorensen, and R. Conradi (2008). Adoption of open source in the software industry. In B. Russo, E. Damiani, S. Hissam, B. Lundell, and G. Succi (Eds.), *Open Source Development, Communities and Quality*, Volume 275 of *IFIP International Federation for Information Processing*, pp. 211–221. Springer Boston.

Heller, M. A. and R. S. Eisenberg (1998, May). Can patents deter innovation? the anticommons in biomedical research. *Science 280*(5364), 698–701.

Hertel, G. (2003, July). Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. *Research Policy 32*(7), 1159–1177.

Huizingh, E. K. R. E. (2010, November). Open innovation: State of the art and future perspectives. *Technovation*.

Jelinek, M. (2004). *Managing as Designing*, Chapter Managing Design, Designing Management. Stanford.

Jorde, T. M. and D. J. Teece (1989, Spring). Competition and cooperation: Striking the right balance. *California Management Review*, 25–37.

Kogut, B. (2000). The network as knowledge: Generative rules and the emergence of structure. *Strategic Management Journal 21*, 405–425.

Kogut, B. and A. Metiu (2001). Open-source software development and distributed innovation. *Oxford Review of Economic Policy 17*(2), 248–264.

Krippendorff, K. (1989). On the essential contexts of artifacts or on the proposition that &quot;Design is making sense (of things)&quot;. *Design Issues 5*(2), 9–39.

Krishnamurthy, S. (2005). *Perspectives on Free and Open Source Software*, Chapter An analysis of open source business models. MIT.

Lakhani, K. R. and E. von Hippel (2003, June). How open source software works: ”free” user-to-user assistance. *Research Policy 32*(6), 923–943.

Lakhani, K. R. and R. G. Wolf (2005). *Perspectives on Free and Open Source Software*, Chapter Why hackers do what they do? Understanding motivation and effort in Free/Open Source Software Projects. MIT.

Lane, P. J. and M. Lubatkin (1998). Relative absorbtive capacity and interorganizational learning. *Strategic Management Journal 19*, 461–477.

Langlois, R. N. (1990). Creating external capabilities: Innovation and vertical disintegration in the microcomputer industry. *Business and Economic History 19*, 93–102.

Leonard-Barton, D. (1992). Core capabilities and core rigidities: A paradox in managing new product development. *Strategic Management Journal 13*, 111–125.

McKusick, M. K. (1999). *Open Sources: Voices from the Open Source Revolution*, Chapter Twenty years of Berkeley Unix: From AT&T-owned to Freely Redistributable. O'Reilly.

Nooteboom, B. (2008). *The Oxford Handbook of Inter-organizational Relations*, Chapter Learning and innovation in inter-organizational relationships. Oxford University Press.

Oba, B. and F. Semercioz (2005, May). Antecedents of trust in industrial districts: an empirical analysis of inter-firm relations in a turkish industrial district. *Entrepreneurship and Regional Development 17*(3), 163–182.

Oliver, A. L. (1994, November). In between markets and hierarchies - networking through the life cycle of new biotechnology firms. Institute for Social Science Research, Working Paper Series issr-1005, Institute for Social Science Research, UCLA. available at http://ideas.repec.org/p/cdl/issres/issr-1005.html.

Pisano, G. (2006, October). Profiting from innovation and the intellectual property revolution. *Research Policy 35*(8), 1122–1130.

Powell, W. W., K. W. Koput, and L. Smith-Doerr (1996, March). Interorganizational collaboration and the locus of innovation:networks of learning in biotechnology. *Administrative Science Quarterly 41*(1), 116–145.

Raymond, E. S. (1999). *Open Sources: Voices from the Open Source Revolution*, Chapter A Brief History of Hackerdom. O'Reilly.

Ring, P. S. and A. H. van de Ven (1994). Developmental processes of cooperative interorganizational relationships. *The Academy of Management Review 19*(1), 90–118.

Samuelson, P. (2006, October). Ibm's pragmatic embrace of open source. *Commun. ACM 49*(10), 21–25.

Teece, D. (1986, December). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy 15*(6), 285–305.

Teece, D. J. (1989). Inter-organizational requirements of the innovation process. *Managerial and Decision Economics Special issue*, 35–42.

von Hippel, E. (2001, Summer). Learning from open source software. *MIT Sloan Management Review*, 82–86.

von Hippel, E. (2005). *Perspectives on Free and Open Source Software*, Chapter Open source software projects as 'user innovation networks'. MIT Press.

von Hippel, E. (2006). *Domocratizing Innovation*. MIT Press.

Watson, R. T., M.-C. Boudreau, P. T. York, M. E. Greiner, and D. Wynn, Jr. (2008, April). The business of open source. *Commun. ACM 51*, 41–46.

West, J. (2003). How open is open enough?melding proprietary and open source platform strategies. *Research Policy 32*, 1259–1285.

West, J. and S. Gallagher (2006). *Open Innovation: Reaching a new paradigm*, Chapter Patterns of Open Innovation in Open Source Software. Oxford University Press.