



# Relationships between open source software companies and communities: Observations from Nordic firms

Linus Dahlander<sup>a,\*</sup>, Mats G. Magnusson<sup>b</sup>

<sup>a</sup> Department of Industrial Dynamics, Chalmers University of Technology, Gothenburg 412 96, Sweden

<sup>b</sup> Department of Innovation Engineering and Management, Chalmers University of Technology, Gothenburg 412 96, Sweden

Received 28 May 2004; received in revised form 22 December 2004; accepted 23 February 2005

Available online 21 March 2005

## Abstract

This paper deals with the relationships between firms and communities in open source software (OSS). A particular feature of OSS is that important resources are not directly controlled by firms, but partly reside within communities that co-exist with the firms. Despite this, firms explicitly try to utilize the resources within these communities in order to create and appropriate value. Consequently, the relationships that firms have to these communities influence their way of doing business. Based on case studies of Nordic OSS firms, a typology consisting of symbiotic, commensalistic, and parasitic approaches to handle the firm–community relationship is developed. Depending on the chosen approach, firms encounter different managerial issues and also use different operational means of subtle control. While firms relying on a symbiotic approach have greater possibility to influence the community through subtle means of control, they are also confronted with more challenging managerial issues. © 2005 Elsevier B.V. All rights reserved.

*Keywords:* Communities; Relationships; Open source software

## 1. Introduction

Open source software (OSS) has lately received much attention from scholars in the fields of organization and innovation. The chief reason for this interest is that OSS challenges substantial parts of the conventional wisdom regarding the role of firms, intellectual property rights, and organizational forms. In particular,

the emergence of firms that deliberately base their existence on the use of software made in OSS communities gives rise to questions, as they explicitly try to utilize communal resources in order to create and appropriate value. Thus, the most striking feature of OSS is that the knowledge to generate software as well as significant parts of the software used in products, without doubt some of the most important resources in this business, is not controlled by the firm, but resides within one or more communities that co-exist with the firm.

While the structures and roles within OSS communities have been researched for quite some time

\* Corresponding author.

*E-mail addresses:* [linus.dahlander@mot.chalmers.se](mailto:linus.dahlander@mot.chalmers.se) (L. Dahlander), [matmag@mot.chalmers.se](mailto:matmag@mot.chalmers.se) (M.G. Magnusson).

(O'Mahony, 2003; Lakhani and von Hippel, 2003; Lee and Cole, 2003), revealing an interesting web of informal structures, roles and relationships, the initially paradoxical existence of OSS firms has received less attention. The OSS movement was founded with the intention to avoid firms appropriating the joint effort of voluntarily contributing developers. Clearly, the challenges involved in exploiting communal resources are significant, leaving many of the OSS firms struggling for survival. Others, however, appear to have found feasible business models. The explanations of the observed differences in performance of OSS firms may depend on a number of factors. It may be that certain firms are better in their exploitation activities, having superior capabilities for marketing, sales and distribution, or they may simply have generated superior products, compared to competitors. Another factor, which can possibly explain why some OSS firms outperform others, is the relationship OSS firms have with the communities that are involved in generating the parts of the software used in the firm's commercial products and services. This leads us to the overall aim of this article, which is to analyze the approaches used by OSS firms to inter-relate to their communities, and to explore the related managerial challenges.

The paper is structured as follows. In Section 2 we discuss the relevant literature of firm and community relations ending with the formulation of three research questions. In Section 3 the method used in the study is explained. Thereafter, the empirical results from four case studies of OSS firms are presented in Section 4. Finally, in Section 5, our findings in relation to the previous literature are discussed and implications for theory and practice are derived.

## 2. Relations between open source software firms and communities

The role of communities that develop innovations and voluntarily diffuse them to their members has lately been observed in diverse fields, such as open source software (Lakhani and von Hippel, 2003; Lee and Cole, 2003), sports equipment (Franke and Shah, 2003; Lüthje, 2004) and librarians (Morrison et al., 2004). Information and support is spread among the people involved in the community; and more impor-

tantly, innovations are shared within this community. Thus, members of the community do not innovate in isolation, but rely on interaction with other members in the community. User communities typically lack financial compensation for those involved, intellectual achievements are attributed to the collective rather than a single actor, and eschew formal planning in the classic sense (Waguespack and Fleming, 2004). The user community literature (Franke and Shah, 2003) mostly focuses on internal mechanisms, such as the rationale for sharing their innovations. The highly related community of practice literature (Brown and Duguid, 1991; Wenger and Snyder, 2000; Wenger, 2001) places its primary focus on learning aspects in occupational and intraorganizational communities. An exception is Lee and Cole (2003), who show how the community represents an alternative to the firm-based knowledge creation. It is evident that relations between firms and communities outside the boundaries of the firms have been less examined, partly because these communities often have an interest in sharing with other members, rather than commercializing their output.

This paper analyzes this issue and argues that firms and communities have divergent rationales for existing, which causes problems when interacting. OSS communities are outside the hierarchical control of the firms, and there are no contractual agreements between the firm and the contributing community members. Even more importantly, the basic idea of exploiting the financial value of jointly developed software runs against the core values of the entire OSS movement, in which the code is protected from being appropriated by commercial firms through the use of legal and normative mechanisms (O'Mahony, 2002, 2003). To illustrate the different rationales that exist between firms and communities, we adopt the taxonomy used by Feller and Fitzgerald (2002) distinguishing between economic, social and technological motivation factors. Table 1 draws up a summary of the discussion below.

### 2.1. The rationale for OSS firms

Firms are driven by maximizing profits in the long run. To do so, they employ various strategies to appropriate returns from their investments. It is often assumed that possible ways of protecting knowledge,

Table 1  
Comparing the rationales for taking part in OSS

Motivation area	Firms	Communities
Economic	Pace development and gain competitiveness West (2003)	Monetary rewards Hertel et al. (2003), Lerner and Tirole (2005)
	Business model taxonomy Raymond (1999b) Cutting costs	Low opportunity cost Lakhani and von Hippel (2003) Signaling incentives Lerner and Tirole (2002)
Social	Sharing codes with community Lerner and Tirole (2002)	Belonging to a community Raymond (1999a)
	Perception that software want to be free Feller and Fitzgerald (2002))	Intellectual challenges Himanen (2001), Raymond (1999a)
Technological	Exploiting feedback Lerner and Tirole (2005))	Altruism or general reciprocity Raymond (1999a)
	Diffusion and win adoption West (2003)	Learning Lakhani and von Hippel (2003)
	Promoting standards	Feedback and help Raymond (1999a) Working with new technologies

Note: This table is heavily influenced by Bonaccorsi and Rossi (2003).

and thereby appropriating future returns, influence the firms' incentive to induce investments in strategic innovation (Liebeskind, 1996). This has resulted in much reasoning of how to make a profitable business built on OSS, where fundamental inputs are not controlled by the firm (see, e.g., Raymond, 1999b; Lerner and Tirole, 2002; McKelvey, 2001; Nilendu and Madanmohan, 2002).<sup>1</sup>

West (2003) recognizes that firms release source codes in order to get their product widely adopted, as a wide adoption increases the likelihood of attracting skilled developers and thereby achieving a higher pace of technological development (Lerner and Tirole, 2002). Widely diffused products can also get different first-mover advantages—such as setting technological standards or attaining a substantial market share. Social motivation factors may also play a role. Firms share their codes with the community, and some firms share important norms and values with the community (Feller and Fitzgerald, 2002).

Nonetheless, economic and technological motivation factors are normally more important for firms than the OSS communities' norms and values. In line with this statement, Bonaccorsi and Rossi (2003) found, in their analysis of 146 firms producing OSS products and

services, that these firms were primarily driven by economic and technological factors, rather than by social factors.

## 2.2. The rationale for OSS communities

Given the non-traditional organizational character of OSS communities, the incentives for taking part in these communities have received due attention from the researchers. Recent empirical work has shown that developers contribute as a result of both extrinsic and intrinsic sources of motivation (Hertel et al., 2003; Hars and Ou, 2002; Lakhani et al., 2002). Consequently, within the growing literature on OSS, several studies have focused on the social motivation factors. Helping others may increase self-esteem, demonstrating technical expertise, earning respect and status, and responding to norms of mutual aid (Himanen, 2001; Raymond, 1999a). This implies that contributors become beneficiaries of the public good because they care about the system as such.

Other work stresses that there is an economic rationale for taking part. Surveys of people working in communities show that many receive financial compensation (Hertel et al., 2003; Hars and Ou, 2002). Moreover, Lerner and Tirole (2002) suggest that signalling incentives matters as the contributors are concerned about future careers. Despite this, it seems plausible to suggest that community members developing OSS software at an overall level are much more likely to

<sup>1</sup> Different business models have been tried, e.g., developing businesses based on packaging, support or service, developing and using complementary proprietary software to varying degrees, embedding it into hardware products, etc.

be driven by social factors than traditionally employed software developers.

### 2.3. *Synthesis and formulation of research questions*

In firms, the relationships between developers and their employers are regulated by contracts. Consequently, these developers employed by firms receive salary and other types of financial compensation. In OSS projects, anyone is free to join and the relations are informal. Whereas firm-based software creation is normally restricted to relations within the firm, OSS developers are not bound to firms but are dispersed in all parts of the world.

The use of communities created or induced by management appears to be a balancing act, where the influence from the firm's side, in terms of the degree of control and the strategic direction that is imposed, is a key issue. With too much control it is questionable whether it will be possible to generate the energy, interest and creativity that is at the core of "naturally" emerged communities. With too little control and direction, however, the effects for the firm may be small, or even counterproductive, in case the community's goals work against the organization. This ought to be even more pronounced in the case of OSS, as management of the firm has no formal influence over the community based on their standing in the firm, and the overall value of openness and sharing prevalent within OSS is apparently conflicting with the firms' ambitions to generate profit. This apparent management challenge has directed our study to a number of research questions focusing on the inter-relationship between the firms and the communities, the first one of which is as follows:

*Research question 1:* What different approaches exist to handle OSS firm–community relationships?

The inter-relationship between OSS firms and communities seems to comprise a set of tensions and inconsistencies in terms of goals, norms and values, potentially leading to different managerial issues. This leads us to the second research question.

*Research question 2:* What managerial challenges do OSS firms encounter in their community-related activities?

Finally, we address the way that OSS firms deal with their communities, at an operational level. Thus, we pose the third research question.

*Research question 3:* What operational means do OSS firms use in order to handle their relationships to communities?

## 3. Method

### 3.1. *Methodological approach*

In order to explore the inter-relationships between OSS firms and communities, in-depth case studies of one Finnish firm and three Swedish OSS firms have been performed. The rationale for studying multiple cases is the need for data regarding different approaches to inter-relating with OSS communities, as well as the perceived difficulties and opportunities involved in using them. Given the explorative nature of the investigation, generalizations are only made with respect to theory.

### 3.2. *Data collection and analysis*

As common in the case study approach, several data collection approaches were used (Maxwell, 1996; Miles and Huberman, 1984), because using several data sources enables triangulation of evidence and construct validity (Yin, 1984). Three primary types of data sources were collected.

1. Secondary resources were gathered on all firms from annual reports, company directories, business and specialist press and homepages. All information was used to get an idea of the competitive environment, important milestones and the perception of outsiders of the firm. It formed a useful background to later steps and provided us with the possibility of comparison with other data sources.
2. Semi-structured face-to-face interviews were carried out at the firms. Two pilot interviews were carried out to learn how to use the interview manual and test the relevance of the questions. Each interview lasted 0.5–3 h and included a comprehensive number of questions about the relationship to the community. At each firm, two to four interviews were carried out depending on the com-

Table 2  
Background information about the cases

	MySQL	Cendio	Roxen	SOT
Founded	1995	1992	1994	1991
Owners	Private, VC	Private, VC	Private, VC	Private, VC
Revenues 2002	5257	857	1217	946
Revenues 2001	514	2433	3301	N/A
Employees 2002	32	19	58	N/A
Employees 2001	12	39	58	N/A
Profits 2002	–1610	–106	–4715	–506
Profits 2001	–884	–372	–3385	N/A

Data from annual reports and other secondary data. All numbers in 1000 €.

plexity of the relationships between the firm and the community, and the amount of information obtained from the interviews. The interviews were recorded and transcribed within 2 days. In three cases, the respondents were reluctant to having the interviews taped, and these interviews were instead documented through extensive and careful note taking. A draft of the empirical observations was sent to the respondents to ensure that we interpreted the details correctly, and for them to give comments upon.

- To further analyze the relationship with the firm's community, we followed mailing-lists and forums at least three times a week over a period of 4 months. This was possible in three of the cases, where there actually was a community-established in relation to the product. This comprehensive study was used to get an idea of how the users and developers in the community act. In total, thousands of emails and conversations were screened.

The different data sources allowed us to form case studies for each one of the individual firms, which thereafter were compared to observe similarities and differences (Eisenhardt, 1989).

The main weakness of the qualitative approach used is that it does not allow for the use of statistical inference. A few general observations about the environment in which these firms act can however be made. Unfortunately, there is currently no comprehensive database on the number of firms in OSS in the Nordic countries, so much effort was devoted to selecting relevant firms through gathering data on all firms found in the business press, homepages, etc. The number of Nordic firms focusing only on developing open source products and services is relatively

low—fewer than 50. Furthermore, several initiatives were not successful. An indication of this is that at least four firms went out of business between June 2003 and May 2004.

### 3.3. Brief description of the firms

We analyze four small, relatively young firms working with OSS. Table 2 shows useful background information. A few points are worth noting with the respective firms to understand the nature of their business and competitive environment.

- MySQL develops a widely used database, originally developed by the founders of the firm. The database is free of charge for leisure, whereas it costs to use for commercial purposes. The cost is, however, considerably lower compared to other databases such as those developed by IBM and Oracle. The community consists of a huge number of users and developers.
- Cendio develops and sells a thin-client (a software framework in which computer terminals use software which runs on a server, rather than from each single computer) which combines in-house development with modules that community-established projects have developed. It is based on bundled software that consists of approximately 80% OSS software from different projects while the remaining 20% is proprietary software developed by the firm. Cendio has not been actively involved in building a community, but uses the work that peer-established communities develop.
- Roxen develops two inter-related products—a web server and a content management system that runs on this particular web server. The web server is licensed under the GPL-license, whereas their add-on product

is proprietary software. The firm founded the community, but it has over time become less central for the firm as many users and developers have moved to other projects. The main OSS competitor is unarguably Apache, but many other firms are developing web-servers.

- SOT has released a Linux desktop, a Linux server and an office suite, which can be downloaded from the community-established by the firm. By giving

away the product for free, SOT tries to build up their reputation as a knowledgeable and trustworthy partner and thereby be able to sell services to customers. The firm helps customers to integrate OSS according to their needs and hence builds different types of solutions.

Table 3 summarizes and illustrates the differences in relationships to the community.

Table 3  
Characteristics of the cases

	MySQL	Cendio	Roxen	SOT
Community design (design, number of users, developers, customers)	Founded by three individuals who previously worked with a database in the community	Founded by people from a university who were devoted to OSS	Founded by computer scientists from a university who commercialized the projects they were working on	Founded by an entrepreneur with great interest in OSS
	MySQL mailing-lists of questions in different categories. Stored on the company's website Over 4 million active MySQL installations worldwide 4000 paying customers	Does not have an established community around the firm	Roxen community for the open source web server with forums, chat, articles, download area, etc. Roxen mailing-lists – development issues	SOT community – simpler questions  SOT mailing-lists – development issues  Users all over the world, but primarily in Finland, the Baltic countries and Russia
Relationships to community	Firm-established community as well as several other forums Dual licensing	No firm-established community  Follows the evolution of different projects that are of relevance and build up internal competence	Firm-established community  Hybrid OSS firm. Develops its web server under the GPL-license, but makes money on an additional product	Firm-established community  GPL-license
	Gives and gets code	The firm gives back codes that are not key for the company's product to relevant projects	Arranges user conferences in real life	Users involved in translations and burn projects
	Bug reports	Deals with licenses that are vital for the company	Gives and gets code	Arranges user conferences in real life
	Arranges user conferences in real life		Bug reports	Gives and gets code  Bug reports

#### 4. Analysis of the cases

Recapitulating the research questions, we in turn discuss: (1) different relationships between firms and communities; (2) managerial challenges for the firms in relation to communities; and (3) operational means used to handle relationships with communities. In answering the first research question we derive three approaches for handling OSS firm–community relationships: (1) symbiotic; (2) commensalistic; and (3) parasitic. In answering the second research question we distinguish several managerial issues in the relationship with communities, and in answering the third research question we find operational means of subtle control used by the firms. We suggest that depending on which approach is used, different managerial issues and possible operational means are present. Table 4 shows our analytical model which summarizes the analysis, with the three approaches on one axis, and the possibility of influencing the community, managerial issues and operational means of subtle control on the other.

#### 4.1. Different approaches to dealing with the OSS firm–community relationship

The different rationales for firms and communities engaging in OSS (see, e.g., Bonaccorsi and Rossi, 2003; West and O’Mahony, 2004), in terms of firms being primarily profit-driven and communities motivated by social factors, give rise to problems in the interaction between these parties. From the empirical observations, we see that firms handle this issue differently, by having different approaches to relate to the communities. We propose a typology of three different basic approaches used by firms to inter-relate to their communities: (1) symbiotic; (2) commensalistic; and (3) parasitic. The different approaches should, however, not be seen as distinctive categories, but rather as steps on a continuum regarding the benefits for the communities deliberately searched by the OSS firms.

The *parasitic approach* implies that the firm only focuses on its own benefits, without taking into account that its actions might harm the community. This is a possible approach that might occur, even though we did

Table 4  
Synthesis and typology of approaches

	Symbiotic (firm gains–community gains)	Commensalistic (firm gains–community indifferent)	Parasitic (firm gains–community loses)
Nature of relationship	Giving something to the community, often through a firm-established community	Search for useful input from the community	Search for useful input without obeying norms, values and rules
Possibility of influencing community	High	Low	None
Managerial challenges	Respect norms and values Obey licenses Resource consumption of developing community	Respect norms and values Obey licenses Getting acceptance of the community for using its resources in commercial applications	Avoiding direct conflicts
Operational means of subtle control	Attracting developers Aligning different interests Resolving ambiguity about control and ownership Devoting personnel to work in communities Creating and maintaining reputation Fringe benefits Interaction tools Selling development tasks	Devoting personnel to work in communities	

not observe it in our cases. An obvious risk related to the commensalistic approach is that, over time, it turns into a parasitic relationship, where the firm comes to be perceived as a negative influence by the community, either in terms of its violation of basic norms, values and principles, or that it is simply perceived as a free rider. It is clear that no OSS firm would deliberately choose a parasitic approach, as causing harm to the community that the firm is feeding upon does not appear to be a sustainable business model. However, given the fundamental differences between different actors' rationales to participate in OSS development, the line between what constitutes a commensalistic and a parasitic approach may be fine, and not always clear.

Three of the observed firms – MySQL, Roxen and SOT – have actively attempted to create a community in relation to their product, but only MySQL has been successful in reaching a large number of users. Roxen encountered fierce competition from Apache, and did not succeed in diffusing their product as widely as desired. Their focus, therefore, gradually shifted towards more traditional software development, as they could not benefit sufficiently from their community. In SOT, the community has a less central role than in MySQL, and it has over time become less important. Cendio, on the other hand, has no firm-related community, but tries to benefit from the development, taking place in different communities without actively influencing them.

The *symbiotic approach* implies that the firm tries to co-develop itself and the community. In the development of both the firm and the community, the effects on the other party are considered when decisions are taken. In order for this to work, it is necessary for firm management to be directly involved in community development, as legitimacy to influence the community can hardly be gained from having a formal role in a firm, but on the status gained in the community, based on its norms and values. One way of viewing this is to consider the community as an extended part of the knowledge base of the OSS firm, however outside of its formal span of control. This approach is similar to the present practice to develop and manage communities of practice within firms (Wenger and Snyder, 2000), presenting a paradox to managers in terms of handling the diametrically opposed needs for openness and control. However, in the case of OSS, this tension is probably even stronger as community members do not necessarily have any formal connection to the firms, but can

disregard their goals and strategies completely in case they are not in line with those of the community. Of the firms studied here, only MySQL can today be regarded as having a clearly symbiotic approach, even though Roxen and SOT initially revealed similar patterns of activities. Over time, the focus on the community at Roxen and SOT decreased, as the firms had problems in appropriating adequate returns. Using the symbiotic approach implies that the firm is focusing on the realization of mutual benefits for both the firm and its community. While the firms' ambition to manage their communities in this case is significantly lower, the norms and values of OSS are respected and taken into consideration by the OSS firms. Some minor influence on the development direction takes place by active participation in various projects, but there is no strategic co-alignment between the firms and the communities. Firms adopting this approach try to benefit from the work not only performed in their related communities but also go to great lengths in their attempts to reciprocate these benefits. One way of doing so is to give internally developed codes to the communities. Another way for the firm to be perceived as useful by the OSS developers is to provide a well-functioning infrastructure that facilitates the performance of different development tasks and allows for stimulating interaction. The background of key individuals within OSS activities has been an important factor for the firms to be perceived as something positive from the perspective of the communities, even though the relationships between these firms and their communities to some extent have deteriorated over time as the firms have become more commercially oriented.

An intermediate way to inter-relate to the community is to use a *commensalistic approach*, i.e. to benefit from the co-existence with another entity while leaving it without harm. The basic idea in this specific context is to thrive on communal resources that are continually replenished, while keeping the direct involvement in the development of these communal resources to a minimum. Cendio is predominantly using a commensalistic approach, even though there are some components that more resemble symbiosis, primarily the giving away of codes that from a business perspective is not absolutely necessary to retain. In this case, as we are dealing with social systems paying much attention to the diffusion and use of knowledge, one aspect that appears to be important to attend to is the legal mechanisms that govern

the software which commensalistic OSS firms use, and how these mechanisms relate to the norms and values of their communities.

#### 4.2. Managerial challenges in community-related activities

From the case studies, we distinguish seven managerial issues that are critical to attend to in relation to the community: (1) respecting the norms and values of the OSS communities; (2) using licenses in a fruitful manner; (3) attracting developers and users; (4) handling the resource consumption related to community development; (5) aligning different interests about the nature of work; (6) resolving ambiguity about control and ownership; (7) getting acceptance for using the community-developed software in commercial applications and avoiding direct conflicts.

First, from the case studies it is clear that a critical challenge is the *norms and values* that defend the communal resource from being depleted by firms. Beside the legal mechanisms (primarily licenses), the joint effort is protected through the social norms and values that are diffused across users and developers. Despite the difficulties to influence the norms and values of communities, some attempts to do so were noted in the case studies. Key individuals within projects have a greater possibility in doing so. In the case of MySQL, the firm emerged as the result of a few people jointly developing a database. These individuals appear, at least to some extent, to have the capacity to influence the community, as they are well-known and respected by it. The firms that have established a community have also been active in creating social events—such as fairs and workshops for users and developers. In a related line of argument, O'Mahony and Ferraro (2004) find that face-to-face interactions are crucial in managing the boundaries of open projects. Apparently, social events are another means of proactively shaping social norms and values, and creating acceptance for the commercial use of knowledge created by the community.

Second, a substantial challenge when shaping the relationships with the OSS movement is to *handle the different licenses* that govern how the software resulting from OSS projects can be used, as the ownership of a project is a central theme in OSS (Raymond, 1999a). Licensing schemes are, therefore, of great importance as they influence how the software ought to be used,

and also have a significant symbolic value (Lerner and Tirole, 2005). Hardly surprisingly, licenses are considered to be extremely important for all firms, and they are influenced by existing ones. They have to cope with the problem of using existing software modules developed by communities. When communities develop their software they protect it from being depleted by firms through reversed copyright schemes. This in turn limits the possibility for firms to use it in conjunction with internally developed source codes.

Third, an obvious challenge for the observed OSS firms is to attract not only customers, but also *developers* that can contribute to the development of new software. Even though numerous studies have emphasized the benefits of OSS in terms of taming complexity (Raymond, 1999a), satisfying heterogeneous user needs (Franke and von Hippel, 2003), and enabling the possibility of bug reporting and development of new codes (Lakhani et al., 2002), the use of OSS does not imply that developers and users automatically get interested in the project and contribute to the software development. Apparently, a vast number of projects compete for the attention and interest of the developers and users. As many developers are motivated by social factors (Raymond, 1999a; Himanen, 2001), firms have to provide stimulating challenges and fun projects for developers and at the same time create products that are simple enough to attract users. At the very same time, firms have to be able to sell customization and other services to their customers.

Fourth, the firms face the issue of *resource consumption related to community development*. In order to create and maintain relations with the community, the firms had to devote considerable resources (time and money). The three firms building communities – MySQL, Roxen and SOT – all made considerable investments in creating their product. For example, the MySQL community was founded after the release of the database, and the firm has since then been active in releasing new improvements and functionalities, devoting resources to building infrastructure and organizing social events for people working in the community.

Fifth, working tightly with the community implies that the firm needs to be able to *align different interests about the nature of work*. The intellectual challenges for community members noted above vis-à-vis the firms carrying through routinized tasks were, for example, noted in the case of Roxen. As the product

developed by the firm matured and the primary focus changed from developing the product together with the community to selling it to customers, the firm found it increasingly difficult to work closely with the community.

Sixth, firms that are active in creating new projects need to *resolve ambiguity about control and ownership*. Earlier studies have shown that ownership of projects is critical within OSS (Raymond, 1999a; O'Mahony, 2002; O'Mahony, 2003; West and O'Mahony, 2004). Consequently, firm involvement to some extent obstructs the possibility for a community to have the desired ownership.<sup>2</sup> Two out of four firms – Roxen and SOT – had experienced problems as the interests of firms and the communities were conflicting in terms of ownership. The developers and users of the Roxen web server went to other projects after the firm released its proprietary add-on. As the firm consciously moved away from the OSS concept, conflicts arose with the users and developers. Cendio is also a type of firm that the OSS movement consciously attempts to hinder from using community-developed codes in other ways than regulated in the licenses. Firms that release a new project – represented in the cases by MySQL, Roxen and SOT – have the possibility of making a choice of which license to use. MySQL over time changed to a dual strategy—using both the most commonly used license (GPL) and a firm specific license in order to make a clear distinction between those users that have to pay and those who can use the product for free. Roxen used the GPL-license for its product, but later on decided to sell commercial licenses for an add-on, implying a deliberate step away from the active use of the community. SOT had an interest in getting their product widely diffused and consequently used a GPL-license.

Seventh, firms that use community-established software need to get acceptance for using the community-developed software in commercial applications and *avoid direct conflicts*. The communities largely depend on innovations being improved and shared with others. Firms like Cendio that use existing modules and combine them in a framework, need to get acceptance from the community that as long as they obey licenses, they are not sharing everything they develop. One way of doing this is through being clear on the pieces of software the firm uses from different communities, and how that

is used in conjunction with internally developed source code. Cendio, for example, takes active part in fairs and forums in the area of OSS.

#### 4.3. *Operational means for handling the relationship to communities*

Firms do not rely on direct control over the developers and users in the OSS movement, as there is no formal relationship between them. Instead, subtle means of control that aim at influencing the community in a certain direction are used. From the case studies, we distinguish five mechanisms through which this can be achieved: (1) devoting personnel to work in or with communities; (2) creating and maintaining reputation; (3) fringe benefits; (4) the use of 'interaction tools'; and (5) 'selling' development tasks.

First, *devoting personnel to work in or with communities* was observed as a means of subtle control, e.g., in the case of Cendio. By working as peers in projects, firm representatives can keep track of the progress and sometimes even influence decisions. Skilled personnel may also gain a good reputation in the eyes of community peers, which in turn gives attention to the firm.

Second, *creating and maintaining reputation* is an important mechanism of subtle control (Raymond, 1999a; Lerner and Tirole, 2002), which serves as a signalling incentive (Holmstrom, 1999). A consequence of this is that firm representatives within OSS who are well-known and respected in the communities, have a higher ability to influence the development activities performed in the community compared to less well-connected ones, something which was especially obvious in the case of MySQL. Reputation is also important from another angle, namely in terms of recognition of skilled individuals, something, which constitutes a motivating factor for individuals to take part. The firms seem to be aware that giving credit to people that help out with bug-hunting, new pieces of code and translations are of vital importance. Consequently, all the observed firms with their own communities made use of this specific control mechanism.

Third, different kinds of *fringe benefits* are used to encourage a certain type of behavior. Again, these are only used when firms have a community-established in close relation to the firms' products. Sometimes the companies devote CDs or computer equipment for a given task. SOT, for example, used this strategy when

<sup>2</sup> We are indebted to an anonymous reviewer for this point.

attracting bug-hunters for their new release. Another kind of fringe benefit is to get access to new versions earlier than others. However, it should be noted that the firms perceived it as extremely hard to create sustained interest by using fringe benefits.

Fourth, the use of ‘*interaction tools*’, when developing software, can serve as a means of subtle control. These tools are services that developers and users might be willing to use (such as on-line forums, mailing-lists, etc.). This is partly related to the notion of innovation tool-kits (von Hippel and Katz, 2002), but does not necessarily have to do with outsourcing need-related innovations to users, but rather in governing the infrastructure in the intersection between the firm and the community. Also, if the interaction tools are well designed, they may form a social function in that they allow for interaction between different developers or between personnel and developers. As mentioned above, the interaction is not necessarily limited to virtual spaces, but can also take the form of organized meetings in the real world. Also at these gatherings, the OSS firms have the opportunity to more or less directly influence the development direction.

Fifth, ‘*selling*’ development tasks is a possible way of influencing a community. Many developers who work in the community are motivated by intellectual challenges (Raymond, 1999a; Hertel et al., 2003; Lakhani et al., 2002). From the perspective of the firms, this implies that the tasks they provide have to be perceived as interesting. MySQL managed to do so by offering an interesting product and constantly improving the product at a rapid pace. SOT and Roxen, on the other hand, experienced difficulties in coming up with challenging tasks. When SOT first released its distribution and office suite it had functionalities and translations that the main competitor in that segment did not have, and, therefore, attracted developers and users. However, as the development tasks over time came to be of a more incremental nature, activity in the community dropped.

## 5. Conclusions and policy implications

The above suggests that OSS firms can use symbiotic, commensalistic, or parasitic approaches for inter-relating to their communities. By using a more symbiotic approach, firms have more possibilities to

influence the community. However, a symbiotic approach implies the acceptance of dual roles, and the key issue becomes how to balance a distributed knowledge system incorporating both the firm and its community, also acknowledging that the modes of control available differ widely within this system. These firms have much larger possibilities to use various operational means of enforcing subtle control. Yet, this is not an easy task, as several managerial issues emerge: (1) respecting the norms and values of the OSS communities; (2) using licenses in a suitable way; (3) attracting developers and users; (4) dealing with the resource consumption involved in community development; (5) aligning different interests about the nature of work; and (6) resolving ambiguity about control and ownership.

The commensalistic approach, principally trying to utilize existing communities without inflicting any harm, may at a first glance appear to be easier to handle, but nevertheless holds a number of potential problems. Firms that are not involved to the same extent mainly face the problem of getting acceptance for using the community-developed software in their business activities and avoiding direct conflicts, but have very limited possibilities of influencing the community. Consequently, firms choosing a commensalistic approach will have to develop a capacity to adapt their strategies not only to provide what the customers want, but also to a significant extent to the development taking place in the communities outside the firm. By not being actively involved in community development, it may be significantly harder to get acceptance for the firms’ commercial use of the communal resources. Hence, there is a greater risk of being perceived as parasitic, leading to the possible deterioration of the relationship.

This typology of approaches to relationships and the underlying managerial issues and operational means of subtle control also have the possibility of explaining the change from one approach to another. It shows that a greater possibility of influencing might result in several benefits, but it also results in a number of managerial issues that we have outlined.

The relationships between firms and communities voluntarily sharing their innovations also have policy implications. The communities analyzed here have evolved due to firm initiation and organizing among peers. People working within the communities voluntarily share their innovations with others, and their achievements are not protected by intellectual property

rights (Waguespack and Fleming, 2004). Our observations indicate that firms may also benefit from this, through creating and maintaining relationships with these communities.

The ‘truth’ of intellectual property rights as *the* answer for spurring economic growth should be taken with great care. The debate in recent years to patent algorithms and business methods related to software has resulted in a heavy debate within Europe, as they discuss the benefits and drawbacks of increasing the possibility to use software patents. It has been noted that strong appropriability regimes may benefit individual firms, but slow the general cumulative advance (Levin et al., 1987). Our paper illustrates that through creating relations with communities firms can create economic impact, which illustrates that firms may benefit from the general advance in communities evolving at a rapid pace. An example of this is the case of MySQL. The firm has in a few years grown to become a major alternative to great software incumbents with millions of installations worldwide. The entire system of activities also includes actors with radically different goals and rationales for existing, and the inherent tensions in this set-up call for new ways of thinking about what a firm should do.

### Acknowledgements

We thank two anonymous referees and the editor Nick von Tunzelmann for comments that improved the first version of this article.

### References

- Bonaccorsi, A., Rossi, C., 2003. Comparing motivations of individual programmers and firms to take part in the open source movement. From community to business. Working paper. Available at: <http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf>.
- Brown, J.S., Duguid, P., 1991. Organizational learning and communities-of-practice: toward a unified view of working, learning, and innovation. *Organization Science* 2 (1), 40–57.
- Eisenhardt, K., 1989. Building theories from case study research. *Academy of Management Review* 14 (4), 532–550.
- Feller, J., Fitzgerald, B., 2002. *Understanding Open Source Software Development*. Addison Wesley, Boston.
- Franke, N., von Hippel, E., 2003. Satisfying heterogeneous user needs via innovation tool-kits: The case of Apache security software. *Research Policy* 32, 1199–1215.
- Franke, N., Shah, S., 2003. How communities support innovative activities: An exploration of assistance and sharing among end-users. *Research Policy* 32, 157–178.
- Hars, A., Ou, S., 2002. Working for free? Motivations for participating in open-source project. *International Journal of Electronic Commerce* 6 (3), 25–39.
- Hertel, G., Niedner, S., Herrmann, S., 2003. Motivation of software developers in open source projects: an internet-based survey of contributors to the Linux kernel. *Research Policy* 32, 1159–1177.
- Himanen, P., 2001. *The Hacker Ethic and the Spirit of the Information Age*. Random House, New York.
- Holmstrom, B., 1999. Managerial incentive problems: a dynamic perspective. *Review of Economic Studies* 66, 169–182.
- Lakhani, K.R., von Hippel, E., 2003. How Open Source Software works: “free” user-to-user assistance. *Research Policy* 32, 923–943.
- Lakhani, K.R., Wolf, B., Bates, J., DiBona, C., 2002. The Boston Consulting Group Hacker Survey. Available at: <http://www.osdn.com/bcg/>.
- Lee, G.K., Cole, R.E., 2003. From a firm-based to a community-based model of knowledge creation: the case of the Linux kernel development. *Organization Science* 14 (6), 633–649.
- Lerner, J., Tirole, J., 2002. Some simple economics of open source. *Journal of Industrial Economics* 52, 197–234.
- Lerner, J., Tirole, J., forthcoming 2005. The scope of open source licensing. *Journal of Law, Economics and Organization*, 21.
- Levin, R.C., Klevorick, A.K., Nelson, R., Winter, S., 1987. Appropriating the returns from industrial research and development. *Brookings Papers on Economic Activity* (3), 783–831.
- Liebeskind, J.P., 1996. Knowledge, strategy, and the theory of the firm. *Strategic Management Journal* 17 (Winter Special Issue), 93–107.
- Lüthje, C., 2004. Characteristics of innovating users in a consumer goods field: An empirical study of sport-related product consumers. *Technovation* 24, 683–695.
- Maxwell, J.A., 1996. *Qualitative Research Design*. Sage Publications, Thousand Oaks, CA.
- McKelvey, M., 2001. The economic dynamics of software: Three competing business models exemplified through Microsoft, Netscape and Linux. *Economics of Innovation and New Technology* 11, 127–164.
- Miles, M.B., Huberman, A.M., 1984. *Qualitative Data Analysis: A Sourcebook of New Methods*. Sage Publications, CA.
- Morrison, P.D., Roberts, J.H., Midgley, D.F., 2004. The nature of lead users and measurement of leading edge status. *Research Policy* 33 (2), 351–362.
- Nilendu, P., Madanmohan, T.R., 2002. Competing on open source: strategies and practise. Working paper. Available at: <http://opensource.mit.edu/papers/madanmohan.pdf>.
- O’Mahony, S., 2002. The emergence of a new commercial actor: community managed software projects. Unpublished Ph.D. dissertation, Stanford University.
- O’Mahony, S., 2003. Guarding the commons: how community managed software projects protect their work. *Research Policy* 32, 1179–1198.

- O'Mahony, S., Ferraro, F., forthcoming 2004. Managing the boundary of an 'Open' project, in: Padget, J., Powell, W. (Eds.), *Market Emergence and Transformation*.
- Raymond, E., 1999a. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, Sebastopol, CA.
- Raymond, E., 1999b. The magic cauldron. Available at: <http://www.catb.org/~esr/writings/magic-cauldron/>.
- von Hippel, E., Katz, R., 2002. Shifting innovation to users via toolkits. *Management Science* 48 (7), 821–833.
- Waguespack, D.M., Fleming, L., 2004. Penguins, camels, and other birds of a feather: The emergence of leaders in open innovation communities. Working paper.
- Wenger, E., 2001. Communities of practice and social learning systems. *Organization* 7 (2), 225–246.
- Wenger, E., Snyder, W.M., 2000. Communities of practice: The organizational frontier. *Harvard Business Review* 78 (1), 139–145.
- West, J., 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32, 1259–1285.
- West, J., O'Mahony, S., 2004. Contrasting community building in sponsored and community founded open source projects. Working Paper. Available at: <http://opensource.mit.edu/papers/westmahony.pdf>.
- Yin, R., 1984. *Case Study Research: Design and Methods*. Sage, Beverly Hill.