ELSEVIER

Long Range Planning 41 (2008) 629–649

LRP long range planning

http://www.elsevier.com/locate/lrp

How do Firms Make Use of Open Source Communities?

Linus Dahlander and Mats Magnusson

Relying on four in-depth case studies of firms involved with open source software, we investigate how firms make use of open source communities, and how that use is associated with their business models. Three themes – *accessing*, *aligning* and *assimilating* – are inductively developed for how the firms relate to the external knowledge created in the communities. For each theme, we make an argument about the tactics associated with each theme and their positive and negative consequences. The findings are related to the literature on the open and distributed nature of innovation, and various theoretical and managerial implications are discussed.

Introduction

A large body of literature argues that innovation processes are often open and distributed and transcend organizational and geographical boundaries.[1] This line of argument indicates that innovation stems not only from a firm's internal investments in R&D, but also relies on important inputs from a variety of external sources, including competitors, consumers, public research institutions, universities and other types of organizations.[2] To access expertise from beyond their boundaries, firms use a variety of mechanisms including licensing agreements, alliances, joint-ventures and informal social interactions.[3] Chesbrough, one of the most frequently cited authors on the topic, notes that '*firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as firms look to advance their technology*'.[4] The distributed nature of the innovation process is particularly evident in instances where new technologies have facilitated the rapid exchange and diffusion of information, with decreased transmission costs and to a larger range of potential participants.

One example of such a distributed innovation process is free and open source software (FOSS), which allows for software code to be continuously improved, modified and diffused. Using communication technologies, participants in FOSS can jointly create advanced software solutions, and new developments are shared in a collective manner within communities, and are often safeguarded from depletion by firms. O'Mahony, for instance, notes how the use of various kinds of legal and normative mechanisms within FOSS communities hinders firms from 'stealing' source code and making it proprietary.[5] Many firms, therefore, face the challenge of extracting profit

0024-6301/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.lrp.2008.09.003

from their engagement in FOSS, as they need to comply with the licenses that govern communities and which make 'derived work' freely available, while at the same time they must defend their own intellectual achievements.[6] While several studies have investigated how firms are trying to do business in FOSS, they have generally focused on what these firms sell and how they use a blend of FOSS and proprietary software.[7] These studies have established that firms try to make profits through re-packaging, supporting, providing services and embedding complementary proprietary software in hardware products.[8] We argue that there is a need for a greater understanding of the different ways that firms work together with communities, as firms' strategic decisions are not taken in isolation from this engagement. In this study we examine *how firms make use of communities* and how this use is associated with the firms' business models. To investigate this research question we use four in-depth case studies of FOSS entrants from Sweden and Finland to explain how these firms changed their business models and their relationships with communities.

This study addresses three themes to probe how firms are using FOSS within their business models: (1) *accessing* communities to extend the resource base; (2) *aligning* the firm's strategy with that of the community; and (3) *assimilating* the work developed within communities in order to integrate and share results. For each of these themes, we outline the tactics used by firms to increase the resource base available for them to build on and to accelerate their technological development. However, using communities increases the risk of losing unique advantages, as important resources cannot be protected in traditional ways, and become more publicly available. Both the communities and FOSS firms undergo changes over time, through a dynamic interplay that influences both the business models in use and the relationships between communities and firms, and which in turn promotes changes in the tactics used by firms to benefit from their involvement with the communities.

The article is structured as follows. The next section presents the theoretical background to the open and distributed nature of FOSS innovation. We review the literature on how firms have attempted to build business models based on FOSS, concluding that the way that firms align their business models with communities has been poorly understood in research terms. The following section describes the research setting and methodology for the study, and explains the rationale behind using four in-depth case studies of firms attempting to commercialise FOSS. The next section describes the findings from this empirical investigation and develops our themes explaining what firms do. The final section discusses the findings in relation to the existing literature, and concludes with some implications for management.

firms benefit from the creative ideas of individuals outside the company but the inflow of such ideas does not happen spontaneously

Making use of communities

A central idea in the growing stream of literature on open and distributed innovation is that firms can benefit from the creative ideas of individuals outside the company. However, the inflow of ideas and innovations from external parties does not happen spontaneously; new strategies and ways of working are required to create a good fit between what the firm does and the resources and capabilities available in its external environment. In this context, FOSS firms are an extreme case, as innovations are developed by communities of distributed individuals, which means that traditional means for handling external input from other companies (e.g. contracts) cannot be applied.

It should be noted that an unrestrained focus on benefiting from external sources of innovation is hardly a fruitful way forward. von Zedtwitz et al. point out that in order to invest in open innovation activities, firms need some degree of control over a number of the elements in their networks.[9] Recent research provides evidence that, up to a certain point, openness is beneficial, but this

benefit decreases where too many external sources of innovation are involved.[10] Yet the research falls short in explaining how external knowledge can be orchestrated in such a way that the firm can be open, but at the same time able to appropriate sufficient returns to survive.

A number of successful FOSS firms have demonstrated that it is possible to rely on communities to provide a fundamental input into product development over a prolonged period, implying a shift from owning important resources to coordinating them. At the same time, we can observe many failed attempts to build businesses on communities, so, while the idea of using communities may be appealing, how to make the relationship work in practice is not obvious. How a firm actually makes use of communities' input influences its ability to build a sustainable business model. This is important, as such work with communities can provide a new way for firms to benefit from innovation processes, which, to a large extent, occur outside their boundaries.

Some research has seen firms' engagement in FOSS communities as rather passive. For instance, an influential paper by Lerner and Tirole argues that firms engage in FOSS for reasons unassociated with their revenue streams, such as public relations and gaining legitimacy, or for learning, and portray the resulting situation as one of passive firms trying to benefit from spillovers.[11]

Others scholars argue that some firms have business models that promote pro-active engagement with communities to develop new products and services.[12] Existing FOSS community values and practices prevent firms from using communities to get unique access in their endeavours to build products and services, and, as a consequence, many have to rely upon new means of generating revenues. Defending the firm's competitive advantage can also be cumbersome, as it is often unable to rely on formal methods of appropriability, such as patents, because the practice in communities prevents firms from making the source code private. Instead, FOSS firms have to build appropriability strategies based on other means of protection.

Several surveys have demonstrated that such formal methods of appropriability as patents and copyright are not a universal solution to the protection of ideas and inventions.[13] Indeed, the results from these surveys suggest that informal methods, such as lead times, network effects, first-mover advantages and access to complementary assets, can be more beneficial for appropriating returns in many industries.[14] This is underscored in the literature on standards, which illustrates how firms can have 'open' strategies and benefit from associated products.[15] The case of VHS vs. Betamax, for instance, exemplifies how the strategic actors behind what, at the outset, appeared to be the weaker performing technology managed to orchestrate a wider array of partners, resulting in increased value for customers and end-users, and, consequently, greater diffusion. Thus openness caused by information being voluntarily (or even unintentionally) divulged to outsiders does not always decrease the chances of being successful. Network effects can be advantageous for those products and services where the benefit that one individual derives increases with the numbers of others, causing a spiral of path-dependent behaviour.[16] By being open towards others partners, for example, through creating alliances or giving away developments for free, firms can appropriate returns from a larger user base and creating lock-ins. Fruitful relationships with communities can therefore be an important mechanism for creating revenues and gaining protection from competitors.

To benefit from external knowledge sources, firms must identify where the knowledge resides and how it can be captured and subsequently used.

To benefit from external sources of knowledge, it is necessary for firms to identify where the knowledge resides, how it can be captured, and how it can subsequently be used. In order to handle this, firms need to develop in-house expertise, or absorptive capacity, to be able to screen and assimilate network developments.[17] While the first ideas concerning absorptive capacity focused

on the development of background knowledge that allows for the identification and evaluation of potentially useful external knowledge, later developments of this concept point to the necessity to include the broader issues of how firms recognize and understand, assimilate and apply external knowledge.[18]

The idea of absorptive capacity demonstrates that the assimilation of external knowledge is partially dependent on the internal knowledge base, as this guides both search behaviours and the possibility of evaluating resources and putting them to use. However, our understanding about how absorptive capacity is manifested in terms of the tactics firms use to access external knowledge, and how these tactics link into their business models, is limited. More specifically, we are interested in how firms make use of external communities as part of their business models. The insights deriving from this should have some theoretical and managerial implications for how firms outside FOSS can manage their 'external interfaces' - by building strategies that combine externally available knowledge with the knowledge residing inside firms in a way that allows them to appropriate the value this creates.

Methodology

Research setting

FOSS provides an interesting context for analysing how firms make use of communities, because firms have emerged that are trying to commercialise software that has been developed in an environment characterized by extreme openness. FOSS development depends on the efforts of a large number of geographically dispersed individuals participating in different communities. This way of working allows more people to be involved in the process of software creation than the individuals within the boundaries of a firm.[19] In addition, FOSS is open and made public; its definition includes statements regarding the right to usage and redistribution, and that it must not restrict other software nor discriminate against fields of endeavour.[20] Community participants protect their joint work from being appropriated by commercial companies through the use of legal mechanisms (licensing schemes and non-profit organizations) and normative mechanisms (punishing those who do not behave in accordance with the norms and values).[21] Consequently, important inputs to the innovation processes reside within communities, and are also to some extent protected from exploitation by firms.

Methodological approach

Firms' engagement in FOSS is a new and emergent phenomenon that is rather poorly understood. We have chosen a qualitative methodological approach using an exploratory multiple case study.[22] Based upon maximum variation logic,[23] we chose to analyse four firms whose common business idea is to commercialise FOSS, but which differ substantially in terms of their products and services, as well as in their business models. The four firms studied here have all been active in trying to commercialise FOSS, but over time have moved in different strategic directions. One major explanation for this change has been due to challenges associated with relying on communities and making use of external knowledge. All firms are dedicated FOSS firms in the sense that they have been attempting to commercialise FOSS as their main business model. (A detailed description of the case selection and methods applied can be found in the Appendix).

We collected and analysed data from a number of different sources for each case. When analysing the data, we first identified a general pattern from examination of the individual cases. After developing thick descriptions of the individual cases, we compared cases. This allowed us to identify some of the different basic tactics employed by firms in their common aim of trying to make use of communities. For each tactic, we noted the positive and negative consequences reported by the firms. We merged the tactics used by the firms into three general themes that formed the basis of our analysis, and used quotes from the interviews to support the analysis of the different tactics. To provide a greater understanding of the firms, Table 1 presents an overview of the firms selected for this study.

Table 1. Firm attributes

	MySQL	Cendio	Roxen	SOT
Founding year	1995	1992	1994	1991
Ownership	Private, VC	Private, VC	Private, VC	Private, VC
Country of origin	Sweden	Sweden	Sweden	Finland
Business model				
Current focus	Develops a widely used database, originally developed by the firm's founders. Applies a dual licensing strategy where the database is free for leisure users, but costs for commercial purposes. Services and training are sold.	Develops a thin client based on bundled software consisting of approximately 80% OSS from different communities, and 20% proprietary software developed by the firm.	Develops 2 inter-related products - a web server and a content management system that runs on this web server. The web server is licensed under the GPL, whereas add-on products are proprietary software.	Releases a Linux desktop, a Linux server and an office suite that can be downloaded from the community established by the firm. Gives away the product for free to gain reputation, and sells integration and customisation
Major changes	Changed licensing strategy from a firm-specific to a dual licensing strategy to more effectively align with community interests.	Changed from mostly selling services related to OSS to creating a product.	Moved away from the OSS idea to become more of a traditional software developer.	Tried selling Linux distributions, but changed to appropriating returns. Focus on selling expertise related to the OSS.
Community approach				
Established a community	Yes. The community consists of a huge number of users and developers.	No. Cendio has not been actively involved in building a community, but uses other communities.	Yes, but less central role today. The firm founded the community, but it has over time become less central for the firm as many individuals have moved to other communities.	Yes, but less central today. The firm founded the community, but it plays a less central role in strategy today as it mainly functions as a marketing device.
Community relationship	- Firm established community as well as several other forums; - Code contributions are assigned to the firm; - Hosts forum and mailing lists;	- No firm established community; - Follows the evolution of different communities whose devlopments can be used in	- Code contributions are assigned to the firm; - Owns the copyright to the web server and the programming language;	- Created incentives by giving away products for free; - Code contributions are assigned to the firm; - Hosts forum and mailing lists;

(*continued on next page*)

Table 1. (*continued*)

	MySQL	Cendio	Roxen	SOT
	- Engages in interactions with users on the webpage, forums, mailing-lists and workshops; - Receives bug reports, suggestions and patches of code from developers; - Active involvement in key aspects of the development, such as standards setting.	conjunction with internally developed source code; - Limited possibilities to influence the direction of development in the communities.	- Difficulty in finding interesting projects to attract the community; - Engage in interactions with users on the webpage, forums and workshops.	- Engage in interactions with users on the webpage, forums, mailing-lists and workshops.
Integrating the work of communities	Code contributed to the firm's product inspected in communities, but also extensive internal analysis.	Exploring the licences for the software that is bundled in the product.	Code contributed to the firm's product inspected in communities, but also extensive internal analysis.	Code contributed to the firm's product inspected in communities, but also extensive internal analysis.

The business models of the cases studied

MySQL

The MySQL database project initially evolved without any explicit intention to commercialise. A few core individuals comprised the group that was active in developing the project. One of the founders, while attending a seminar, met up with a key person from the open source movement, and the idea captivated him to the extent that he decided to adopt the FOSS way of working for MySQL.

Following some hectic years of development, the MySQL database rapidly spread world-wide as it provided efficient solutions and cut costs by as much as 90% compared to competitors' products. The company systematically built and maintained a community around its product by providing support, opportunities to interact via mailing lists and other forums, and other tools that helped community participants in using the database. Thousands of users subscribe to the firm's mailing lists, where experiences are shared and problems discussed and solved. MySQL also actively nurtured the relationship by releasing source code to the community, and the firm has also adopted the practice of allowing individuals in the community as well as paying customers to submit bug-reports and code patches. Code contributions are controlled and evaluated by in-house software engineers before they are included in the product. A certain degree of peer-review also takes place in the community, where the code is evaluated and discussed.

MySQL decided to give the product away for free, [to achieve] a rapid and widespread diffusion of the database

MySQL decided to give its product away for free. According to its managers, the firm aspires to achieving a position where it can sell services and education as well as increase its paying customers numbers, and they reasoned that rapid and widespread diffusion of the database was vital to achieving this objective. The distinction between those that can use the database for free and those that have to pay has caused some problems, resulting in changes to the licensing strategy to cope with the problem of being open, while retaining the ability for the firm to make money. Previously, MySQL used a Lesser General Public License (LGPL), a modified version of the General Public Licence (GPL).[24] LGPL permits firms to combine OS software with proprietary programs: paying this license fee allowed them to avoid disclosing some of their software and thus keeping it proprietary. The move away from the LGPL to a dual licensing model was designed to help MySQL differentiate more easily between a proprietary user, who needed to buy a commercial license, and a free software user, who could use the database under a general public license. MySQL's current dual licensing strategy includes both firm-specific commercial licences and the ordinary GPL. The firm believed the GPL was being abused by people who were distributing the MySQL server tightly coupled with a proprietary application, claiming that the client libraries were free under the GPL. This dual licensing model differs from a pure free software model, and means that community participants cannot easily 'fork' (start competing projects) because the firm retains the copyright and some degree of control. Contributions from the community are checked and rewritten by company employees to ensure copyright is not infringed. But from the perspective of individuals using MySQL to build applications, this change in licensing policy has had only limited effects: they are still free to use, modify and distribute the database in accordance with the GPL.

Cendio

Cendio has attempted to follow the evolution of different communities but without taking an active part in community building. When developing new products and services, the firm screens various communities. If any of their projects are seen as relevant to the firm's new product, their licences are screened, and an evaluation is made as to whether they can be legally used with the company's product. Cendio then 'bundles' different modules of software in a framework which also includes

internally developed source code for sale to customers. Interviews suggested that the chief motivation for Cendio utilizing existing communities is minimizing the costs of research and development (R&D) by relying on external resources as inputs to the firm's internal innovation processes. Relations between the firm and the communities are not reciprocal - the firm merely screens and assimilates the work of relevant communities. However, software that is not considered crucial to the framework is given back to the community, which to some extent legitimises the firm's involvement.

As Cendio has not established its own community, it is not possible for it to develop a large user base to speed up technological development. It is necessary for community software to be evaluated carefully before it can be included in a product. These evaluations take the form of external peer-review involving FOSS communities, and quality testing within the firm.

Cendio did not established its own community, [so lacked] a large user base to speed up technological development.

This approach has its problems. FOSS communities have an interest in how the products of their work are used, and try to protect their efforts from being exploited by commercial actors. Firms, therefore, must manage to obtain benefit from co-existence with the communities while not interfering with them. For example, the legal mechanisms (such as non-profit organizations, licensing, etc) governing the software used by the firm are an important aspect, as are how these mechanisms relate to the norms and values of the communities. Scrutinizing the licences that govern the different modules to be used is critical in order not to upset the FOSS communities. As one manager noted, an extreme case would be that the legal mechanisms used by communities could force the firm to reveal associated developments that could then leak to competitors.

Another constraint is that development within FOSS projects is not always directly relevant to the needs of the company. As the managers noted, guiding and steering development in a particular direction is extremely difficult when there is no formal contract between the parties. Cendio's employees work on some community projects, both professionally and in their spare time, with the idea that by taking part in these projects they may be able to influence the direction of their development.

Roxen

Roxen originally emerged from a group of students working on two related projects: the development of a programming language, and a web server built on that language. Initially, the firm was strongly in favour of FOSS, but gradually turned towards a more traditional approach. Roxen is now a hybrid FOSS firm, combining FOSS with proprietary software.

Roxen started its own community built around interest in its product, which played a more central role in the company's early years, when the firm's strategy was based on the conviction that giving away a product in order to receive user feedback would enable it to make money. However, establishing a viable business model proved more difficult than expected, and after concerns were raised that this approach brought more problems than benefits, the company's business model became more proprietary. Managers struggled with the dilemma of being sufficiently open to attract users by offering challenging tasks, and needing to perform the less exciting activities necessary to create commercially viable products that would generate revenues. The community still exists, but its numbers have continually decreased to the point where Roxen now has far fewer people involved in its community than does MySQL.

Roxen's web server was a niche product at the time and attracted a community, but it faced fierce competition… as the market exploded

Roxen's web server was a niche product at the time and was able to attract a community. As the number of server installations exploded, the potential market grew. The firm faced fierce competition from other solutions (in particular the Apache web server) that grew very rapidly by offering competitive functionalities developed within a lively community. Over time, Roxen lost market share in the web server segment, and implemented more and more strategies that were characteristic of proprietary software businesses.

Roxen still influences the community in a number of ways, even though its importance has declined significantly in recent years. The rationale behind being open and aligning with the community was to attract users, pace technological development, and establish a standard or achieve first-mover advantages vis-à-vis competitors. The firm was forced to change its approach when it proved difficult to find a viable revenue model. Roxen has gradually moved away from the FOSS concept towards a more traditional approach of selling proprietary software in order to survive.

SOT

In 2001, SOT developed and released a Linux distribution, a Linux server and an office suite. SOT developed a geographic market focus that primarily targeting the Finnish and Russian markets by spending resources on translating into these languages. Although SOT founded a community around its products that attracted some interest, the firm had problems building a business model around it, and also in aligning their strategy with the needs of the community. For instance, SOT experienced a mismatch between the firm's and the community's interests that had direct implications for the business model it was attempting to pursue. SOT wanted to generate greater revenue streams by selling extra functionalities that required a specific payment. To get paid for this extra service, the firm tried to use a registration key mechanism for the special release of its software. Community members argued that this was not compatible with established license practices and the open source ethos. Following massive criticism from its community members, SOT adapted its business model and dispensed with the registration key.

SOT made a major change of strategy and decided to give the product they had developed away for free as a marketing device

Initially, SOT's product filled a niche that was not served by competitors. Its community generated some interest, but was never vibrant enough to speed up the technological development. The community provided few source code contributions, and the firm had to spend significant resources to develop new versions of their product. As a response to the relatively modest interest in the community and the problem of selling the product, SOT made a major change of strategy. The company decided to use the product they had developed as a marketing device by giving it away for free. The current focus is oriented primarily to selling services such as Linux integration and customisation to their customers, but following this change, its community has become even less active. Instead of trying to build up its own community, SOT has changed its model to building up internal competencies in customizing different FOSS solutions for its customers.

Accessing, aligning and assimilating

Based on the case studies described above, we inductively derived three themes that appear to be important for how firms make use of FOSS communities: (1) *accessing* communities to extend the resource base; (2) *aligning* firm strategies with the community; and (3) *assimilating* communities in order to integrate and share results. The analysis and discussion are constructed around these three themes.

Accessing – extending the resource-base of the firm

Firms working with open source software use different approaches to access developments in communities to extend their resource base. We identified two major tactics used: (1) establishing new communities to attract outsiders to work in the firm's area, and (2) identifying and using developments in existing communities. For each tactic, we organised the advantages and disadvantages as noted by the firms: Table 2 provides an overview of the tactics with a brief explanation and the pros and cons of each, illustrated by quotes from the interview transcripts.

A first tactic involves *establishing new communities*. MySQL, Roxen and SOT all tried to use this tactic to get access to a wider array of resources: by contrast, Cendio has never employed it. An advantage of establishing a community is that it creates a platform for attracting outsiders – mostly hobbyists, but also employees at other companies – to work on an area of importance to the firm. MySQL, for instance, has been able to build a worldwide community that attracts many thousands of individuals who use and modify their software.

The firms that used the tactic of establishing new communities worked proactively to shape their niche in the market. When there is a larger potential market, other firms obviously have an incentive to move in. Companies stressed that, rather than move into established mass-markets, they tried to identify and occupy a niche where they could be the dominant player, and then expand this niche by working with the community to develop something unique. MySQL managed to develop their database from a niche product into a commonly used solution and succeeded in fending off competitors. Roxen, in contrast, had the potential to benefit when the numbers of its web server community skyrocketed, but faced fierce competition from other communities, and chose to change its strategy to become more detached from its community. SOT had similar problems in embarking on establishing a community aimed at developing its Linux desktop and office suit in east European languages. While it was initially able to attract some outsiders, the community never reached a critical mass and gradually became less important to the firm.

The companies all faced problems in attracting [enough] outsiders to work in their community to create a critical development mass.

The companies all faced problems in attracting outsiders to work in their community. In a very specific niche, the number of individuals that consider the problem relevant and have the skills to contribute to the development is often too small to create a critical development mass. MySQL managed to establish itself as a major player when their initial niche market turned into a much larger market, and the company benefited from a large and active community contributing to the rapid development of its product and thus increasing its competitiveness. One explanation for MySQL's success is that it offers products that are sufficiently customized to attract a broad set of customers, i.e. it focused on mass-customisation. It should be noted that the capacity for customisation resides to a considerable extent in its active community, which develops new software sequences that contribute to the customisation of the product. Finding a niche for the development of a unique product can raise problems, in that the size of the presumed user base is reduced as products become more specialized.

Another reason for establishing a new community is that it can function as a marketing tool to increase brand recognition – both SOT and MySQL argued that this awareness-raising tactic increased their chances of being recognized by the wider environment. Giving software away and building a community both raise awareness of the brand: firms that can build communities involving large numbers of participants, however, raised awareness in a more substantial way than those who just gave their software away. MySQL's database product is installed on more than a million computers – but their mailing lists of users who have solved problems or reported bugs is only a couple of thousand strong.

Establishing a community carries high up-front costs. Roxen, SOT and MySQL noted that it takes time to develop source code internally and it is often necessary to allocate company

Table 2. Accessing – extending the resource-base of the firm

Tactics used	Explanation	Pros	Cons
Establish new communities	Developing a *de novo* community by releasing source code and establishing community. '*When we began with the project, there were no really useful databases with open source licenses. We began building on the project and others became interested.*' MySQL **S, I**	(1) Possible to attract outsiders to work on the firm's product. '*We get feedback and bug reports from people using our database around the globe*' MySQL **S, I** (2)Dynamically change a niche to a mass-market. '*The number of server installations exploded, and our niche became a huge potential market. This resulted in fierce competition from other solutions. If we would have gotten things more right, we could have profited from it*' Roxen **I** (3) Marketing tool to increase brand recognition. '*Open source is a powerful marketing tool for small companies*' MySQL **S, I** (4) More control in the formative stage of the community. '*There were only a small number of people in the community back then: employees, some people at the university, and some enthusiasts at other place.*' Roxen **I**	(1) Difficulties in managing the niche and competition from other communities. '*Establishing communities is extremely cumbersome. You have to do everything right and attract an audience. Finding bugs and developing new functionalities is difficult in some niches. It is sometimes extremely difficult for users to interpret the code and even tougher to develop – it is just a handful of individuals that have that these particular competences.*' SOT **I** (2) Large up front investment needed. '*It takes substantial effort and resources to build a new community you know – it's been hard to justify that for us. We have used alternative paths to get access to communities*' Cendio **S, I**
Identify and use existing communities	Using existing communities established by other firms or individuals. '*We have been searching widely for ideas, trying to find projects with commercial potential using the expertise that we have acquired. […] Our idea has never been to establish new communities, but to use the expertise that is already out there*' Cendio **S, I**	(1) Building legitimacy in communities that have reached a certain threshold. '*Far from all communities succeed, and we can use the advantage of looking at what succeeds and then try to make use of it*'. Cendio **I** (2) Flexibility in accessing different communities as business models are changed. '*We follow the evolution of different communities and decide where to build competencies that can be used together with internally developed source code.*' SOT **I**	(1) Less prospects for control compared to when establishing communities. '*We have virtually no possibility in influencing communities that we haven't established.*' SOT **I** (2) Uncertainty about which communities are of strategic importance. '*A vital strategic choice for us is the communities in which to build up competences. Several hundreds are founded – but few succeed. […] We therefore screen communities of importance regularly. Are they reasonable or not? Can they benefit us*?' Cendio **S, I**

Note: **I**: Supported by interviews; **S**: supported by secondary sources.

individuals' time to working with the community to spur development. These costs were part of the reason behind Cendio's choice of strategies associated with the second tactic.

A second tactic is *identifying and using existing communities*. These communities can be developed by other firms – or more often – by individuals work on their spare time. Cendio made this a key strategic priority, but it was obviously also available to other firms: SOT, for example, screened other communities that they could use when developing services and customisation for their customers.

This tactic offers greater flexibility than establishing new communities. If the business model changes over time, the firm can access different communities to access new knowledge. When SOT changed their business model after having troubles with profiting from their own community, they put a greater emphasis on using other existing communities. Another advantage of this approach is that it allows firms to choose to work with those communities that have already managed to build a critical mass. Cendio pointed out that thousands of communities are established every year, yet there are very few that become widely adopted and of commercial viability.

This tactic also has its disadvantages. First, it is very difficult to know which communities are of strategic importance. Second (as noted, for instance, by SOT), there are fewer opportunities to influence existing communities than when the company establishes its own new community.

It is also worth noting that both tactics had their licensing problems. Firms who started up their own communities had greater initial troubles in setting licensing policies that fitted both the interests of their firms and their communities. On the other hand firms that sought to work with independent communities often struggled to figure out what the communities' licensing policies actually meant in practice and whether they were compatible with their business model. So such firms tended to work only with communities that used well-established licensing practices to minimize their risk of legal consequences.

Aligning – connecting the firm's strategy with the community

A great challenge for firms working in open source is to align their strategy with the work in the community, as firms and community participants are driven by different motives. Firms need to make profits, and usually have a motive to exclude competitors from using the software, whereas community participants want source code to remain free and publicly available. We identified two major tactics to align firm and community strategies: Table 3 provides an overview and brief explanation of these tactics, together with notes about and the pros and cons of each tactic.

A first tactic is *adopting licensing practices to clarify ownership* of product developed by the firm and its community, as opposed to integrated source code developed by other communities. In so doing, firms can make sure to be a seen as legitimate in the eyes of community participants. In part, this is simply a legal requirement as stipulated by community software licenses. However, as the table show, there are variations in how firms do.

Being specific about licensing practices is a prerequisite for firms to be trusted in the community. If a firm is too ambiguous in this area, community participants may fear that the firm will close the source code and make it proprietary. MySQL underscored the importance of this in being able to attract outsiders to work in the community. Cendio, which only identifies and uses existing communities, confirmed this point, stressing the importance of clarifying the licenses of any community source code they included in their products, which is a legal requirement stipulated by most communities' licenses. Cendio also pointed out that analysing the legal consequences of different licensing practices was time-consuming, and using many different communities with different licensing practices create even more work. The firm's strategic choices therefore include not using too many communities, and making sure their licenses are compatible with what they try to produce. In fact, source code that could be of great relevance to the firm sometimes has to be ignored because of the difficulty of understanding the potential licensing consequences for the firm.

All the firms had encountered obstacles associated with open source licensing practices.

Table 3. Aligning—aligning the firm's strategy with the community

Tactics used	Explanation	Pros	Cons
Adopting licensing practices that clarifies ownership	Developing practices about copyright issues for (1) firm's own products and (2) integrated source code developed by other communities. '*Licensing is one of the absolutely most important issues in open source*' SOT **S, I**	(1) Establishes a basis for collaboration between the firm and the community. '*Being clear of how we do business is necessary to create a 'ring of trust' with the community.*' MySQL **I** (2) Avoid direct conflicts with community participants. '*We try to be explicit on what we do with the source code and make sure that we follow the licenses.*' Cendio **S, I**	(1) Difficult to impose practices that are too proprietary as they upset community participants. '*People working in the community got really upset about this [change in licensing to include a registration procedure]. We were more or less forced to change*' SOT **S, I** (2) Necessary to experiment, but too many changes create uncertainty for community participants. '*We previously used a special type of license which caused uncertainty about when payment was required. This created frustration and we change to a dual licensing model*' MySQL **S, I** (3) Time consuming to analyse legal consequences of licensing practices. '*We have to carefully track the licenses of the pieces of software that we integrate here. It takes time, but we have to minimize the risks of being sued*' Cendio **I**
Influencing direction of development	Influence the direction of development by creating incentive structures for individuals working in the community to work on tasks that are of relevance for the firm. '*Steering the development in a certain direction is extremely difficult, but if a firm succeeds in doing so they can benefit a lot. […] It can be done through arranging competitions, giving donations or financial compensation, or perhaps being active at a higher level by influencing the standards setting.*' SOT **I**	(1) Pace technological development by providing different means of subtle control of community participants. '*Stimulating the community to develop new ideas can be done in many ways. We want them to be active. Some of the things they develop can be valuable for us*' SOT **S, I** (2) Receive feedback and tests from individuals that have used the product. '*I think it is a great advantage to use OSS because it allows people to screen the source code. People working as volunteers in the community can make changes and report things that do not work.*' MySQL **S, I**	(1) Interference with the work of community participants can disrupt the community's way of organizing. '*The OSS developers are exactly like cats. Cats do things out of self-interest — and so do the developers. They can do stuff if they are interested in the same things as the firm. But if they feel bounded and confined or that a firm takes what they have done without giving recognition and abiding by the 'rules' of the community — they are not interested*!' Cendio **S, I** (2) Hard to motivate individuals to work on uninteresting tasks. '*Our community consists of enthusiasts world-wide, but the size is considerably smaller today than previously. The lion's share of the developers has moved to other communities which are 'hotter', and 'cooler' to work with. We were unable to provide stimulating challenges.*' Roxen **S, I**

Note: **I**: Supported by interviews; **S**: supported by secondary sources.

All the firms had encountered obstacles associated with open source licensing practices. As noted above, SOT's experimental license procedure which allowed only paying customers to use the full functionality of its software led to a major clash with its own community, and the firm was obliged to give in to community pressure and remove the restriction. MySQL's problems with the Lesser General Public License, which made it difficult to differentiate when a user had to pay for the license or not, resulted in discussions with community participants interested in the database's licensing policies. Following this dialogue, and taking its strategic priorities into account, MySQL changed to a dual licensing model.

A second tactic is *influencing the direction of development in communities*. The firms cannot impose direct control over communities' activities or priorities, but have to use other practices to encourage individuals in the communities to work on tasks that are relevant to the firm's strategies. Relying on community participants, with whom there are no contractual agreements, is a challenge to management: there are no formal means of controlling these individuals, yet firms need to try to shape the direction of communities' activities. One individual at Cendio likened managing individuals in open source communities to herding cats. However, the potential upside can be substantial — both in terms of accessing expertise and in terms of savings on firms' own development costs — although it is impossible to rely on traditional ways of leading and controlling the development processes.

Rather than trying to enforce direct control, firms used subtler means to steer communities in particular directions, and tried to influence community developments by offering incentives to key individuals in the form of payments or fringe benefits for certain tasks, or even salaries to work in leading community roles. Despite the fact that compensation and incentives would seem to go against norms of open source communities, they are indeed used by several firms to influence the direction of development. Firms are aware of the problems involved in using such methods to gain influence, but they are outweighed by the perceived benefits.

Developing software products and services involves both the creation of novel solutions and more routine tasks such as testing and 'bug-hunting'. Firms noted that many individuals working in communities are stimulated by intellectual challenges, and that the development of new functionalities is normally being seen as a more satisfying challenge than involvement in routine tasks. Acknowledging this preference, firms have tried to make tasks more interesting. For example, SOT experimented by giving away early versions of products to give community participants something novel to work on. To encourage the activity in the community and have more control, MySQL performed many of the necessary routine tasks in-house, e.g. checking code quality, testing functionalities and developing incremental changes. This allowed the firm to handle these tasks efficiently, while simultaneously trying to apply the variety of creative solutions that were being developed in the community. Not all firms succeed in influencing communities, however. Both SOT and Roxen have become more detached from their communities over time, because the cost of exercising this form of influence is simply too great compared to the outcomes. A prerequisite for this way of working is to have firm representatives in key positions within the community with the necessary status to steer the direction of the community's work.

Assimilating: integrating and sharing results

Firms that have managed to access and align with communities have still faced obstacles associated with assimilating the work of the communities. In our analyses, we identified two major tactics used by the firms: once again, Table 4 provides an overview, an explanation and the pros and cons of each.

A first tactic is *devoting resources to evaluate and select source code from communities.* Working with communities creates a wealth of resources for firms to draw on, but this very wealth creates the challenge of selecting what to integrate into the firm's internal processes. Firms coped with this challenge by devoting personnel and resources to evaluate source code submitted by community individuals before integration.

Cendio's problems were related to deciding which communities they needed to build competences in. Where managers decided a community was relevant, Cendio adopted practices for evaluating source code before its integration into their product, looking at the suitability of the source code

Table 4. Assimilating - integrating and sharing results

Tactics used	Explanation	Pros	Cons
Devoting resources to evaluate and select source code	Devote time and resources to screen, evaluate and test source code to improve the integration inside the firm. '*To some extent it is true that 'all bugs are shallow given enough eyeballs — the community develops a number of improvement and new suggestions. The issue is to select what to focus on.*' MySQL **S, I**	(1) Increase the acceptance of employees within the organization. '*I firmly believe that R&D must concentrate on developing processes, quality assurance, software engineering, methodology that fits open source development models.*' SOT **I** (2) Task partitioning — community participants can work on intellectual challenges and employees on less stimulating tasks, such as bug hunting and efficiency tests. '*We've moved to be really traditional in product development that occurs here in this building. Occasionally we get feedback and ideas of what to include in future releases, but our employees carefully analyze what is feasible and make the final decisions*' Roxen **S, I**	(1) Difficult to know where to specialize. '*The key for us is to know what's going on in different communities in order to assimilate that information and use it in our products. We follow a massive number of communities and what they develop, but only have the resources to build competences to use them in a handful.*' SOT **S, I** (2) Time consuming to evaluate source code. '*We have employees control the source code before it is included in future releases. That's our way of having some degree of control of the product*' MySQL S, I
Feeding back non-strategic source code to communities	Selectively share source code developed by the firm back to communities. '*We try to give back source code to communities when we can*' Cendio **S, I**	(1) Builds legitimacy in the community. '*I really don't know what we get back, but it's our way of giving something back to the community I think it something one should do.*' Cendio **I**	(1) Indirectly provide information to competitors about developments within the firm. '*We cannot exclude anyone from using source code that we provide back — but that's part of the game*'. SOT **S, I**

Note: **I**: Supported by interviews; **S**: supported by secondary sources.

and checking various licensing issues to ensure that they were compatible with the product. MySQL had similar problems about where to specialize, and suggestions and ideas from community participants sometimes created alternative development paths that the firm had to choose between.

By devoting employees' time to evaluating and selecting source code, some firms tried to establish a clearer division of labour between the firm and the community. Employees would then focus on tasks that few individuals in the community were interested in, such as bug hunting and efficiency tests. MySQL adopted practices that were directed towards decisions about future releases. Contributions from the community were checked, and rewritten by company employees, to avoid copyright complications. But this did not affect participants in the community using MySQL to build applications, as they were still free to use, modify and distribute the database.

Not all firms managed to get substantial resources to select between from the community they had established. Roxen struggled to find something to use, as its community participants only provided occasional inputs for the firm. This raised concerns among the managers about the viability

of their strategy, and prompted the move towards a more proprietary strategy. SOT had similar problems, and responded by putting greater emphasis on screening other communities.

A second tactic is *feeding back non-strategic source code developed within the firm to communities*. Companies noted that not all code they produced was going to be of strategic importance to them. Where this was the case, rather than holding on to source code they were not going to use, they selectively gave it away to the communities.

The main advantage of this approach is in gaining legitimacy in the community. Source code from the firm can also spur the general development of the community. Cendio, for instance, reported that they attempted to feed back all source code that was not seen as vital for the framework they were developing. This was largely driven by belief in the idea of open source communities: as one individual noted, it was their way of '*giving something back*'. SOT submitted that community license provisions sometimes meant they were forced to release source code back to the community, even if they had spent firm time in developing it. Firms are aware that such situations may arise when using community source code and see it as 'part of the game'.

firms are sometimes forced to release source code they have developed back to the community… they see it as 'part of the game'.

Revealing source code brings about a number of potential issues. The obvious negative consequence may be that competitors benefit by using source code developed by the firm. However, this was a limited risk, as only source code considered to be non-vital was given away to the community. Taken together, the managers suggested that, as long as their firm profited more than their competitors from their interactions with the communities, they had an incentive to reveal their source code to foster cumulative developments.

Discussion

We began this study by arguing that FOSS firms' business models are contingent upon their ability to make use of FOSS communities. Based on case studies, we identified three means by which firms exploit communities: (1) *accessing* communities to extend the resource base; (2) *aligning* firm strategies with the community; and (3) *assimilating* communities in order to integrate and share results. For each theme, we identified some tactics used by firms and their positive and negative consequences.

Theoretical implications

It has increasingly been argued that firms can benefit from getting access to knowledge from actors in the wider environment, and that sharing knowledge can be positive in establishing standards and creating network externalities:[25] yet there is limited understanding about how this comes about. In particular, this study has implications for a number of management topics, including coordination and control and division of labour.

What regards to coordination and control, it is clear that FOSS firms face significant challenges, as they are not only working in a clearly distributed system, but in one where the traditional means of control are of limited use. Consequently, the alignment of development activities taking place inside FOSS firms and in the communities poses new challenges. Some firms simply used an adaptive approach, and did not try to change the direction of the development in the communities in any substantial way, but instead focused on using what was developed in the communities and put effort into integrating this work with internally developed components. Others tried to influence community developments by offering incentives to individuals to work with the community, to enhance both its and the firm's reputation, and give the firm more scope for controlling the direction

of community developments.[26] By virtue of their positions in the communities, individuals sponsored by firms can help align the firm's strategies through their knowledge of new developments in the community and their ability to influence them. For the firm, this type of effort requires allocation of resources to management of both the firm *and* the community. This is not straightforward, as the bases for leading positions in the firm and in the community are different, and the individuals that take on these dual roles must have a very broad set of knowledge and skills.

There is a clear limit to the usefulness of direct command in a truly distributed knowledge system,[27] where the individual often has a higher level of task-relevant knowledge than his or her supervisor/manager, and this phenomenon is even more pronounced in the specific case of FOSS development. The means for achieving influence within and outside the company differ; it is therefore necessary for the company to embrace a whole new set of subtle control techniques not based on hierarchical positions, but to be able to motivate community members by providing intellectual challenges or some form of monetary reward.

some FOSS firms have established a de facto *division of labour … the community gets to do the creative work and the company handles the more routine tasks internally.*

Primarily, influencing communities by providing intellectual challenges and stimulating tasks has its problems because developing software also requires the performance of less exciting, routine tasks. Given the limited possibilities of relying on hierarchical control or contractual agreements for the allocation of tasks when dealing with community members, some FOSS firms have established a *de facto* division of labour, where the community gets to do the creative work and the company handles the more routine tasks internally. In other cases, firms take over developments of less exciting modules that are of little interest to community members. While such a clear split between exploration and exploitation activities is easier for the firm to handle, it would arguably lower its overall performance potential, as it may limit the possibilities of finding and exploiting synergies that might exist in the larger system constituted by firms and communities together. Some firms may be able to overcome the division of labour by building large communities and user tools which facilitate contributions from users in the form of suggestions and improvements, at low cost. Once again, it must be stressed that such situations do not come about automatically; they are based on working in new ways, such as assuming an active role in the FOSS community and using new management means of influence and control. While such an extension of the management boundaries holds the potential to create a virtuous development cycle, it also requires managers to cope with the inherent tensions between the different parties involved.

Managerial implications

The cases in this study were definitely not examples of unambiguous success; all demonstrated a variety of problems involved in making use of FOSS communities. The firms adopted different strategies and changed quite dramatically over time. Drawing on these examples, we can derive some implications for managers trying to harness online communities.

The cases illustrate how firms found it necessary to change their business models to align with the communities. Some firms needed to build a sufficiently large community to create a virtuous development cycle. However, the building and development of communities is not straightforward — just establishing a community does not mean that individuals will necessarily be attracted to becoming members, or that their interest will be sustained over time. Two of the case firms found it difficult to motivate community participants over time, mainly in terms of finding interesting tasks for community participants, and activity in their communities faded away. The nature of the competition also differed among firms, from competition at product level to fierce competition between

communities vying with each other to enrol bright-minded individuals. If a community cannot attract new talent, it is difficult to keep it vital: the development of new functionalities and ideas slows down and the community begins to fade away, which would again necessitate changes in the firms' business models.

Our study underscores the need for managers to go beyond conventional management techniques to influence communities. Firms have to make strategic decisions about what will be undertaken in-house, and when to rely on external sources. Even in the case of MySQL, with a large and active community, there was a sizeable group of in-house employees engaged in evaluating, testing and developing source code. As community participation is based on motivations other than employment in a firm, community members are often not interested in the less exciting types of development tasks, but the firm needs to acquire and retain some degree of influence or control over future developments.

firms [need] to develop sufficient absorptive capacity to benefit from external developments — not only to be able to identify useful external knowledge, but also to assimilate and apply it

When there is high reliance on external actors, firms need to have the competencies and selection mechanisms to decide which developments are critical. Otherwise, there is a risk that the firm will be submerged under a plethora of ideas and suggestions, without the capability to prioritise among them and turn this knowledge into successful products. The reverse can also occur: if too little attention is paid to external actors, firms run the risk of missing out to competitors on critical developments.[28] As Cohen and Levinthal have pointed out, this emphasizes the need for firms to develop sufficient absorptive capacity to be able to benefit from external developments. It should be stressed that this does not mean that firms need only to develop the ability to identify useful external knowledge; they must also be able to assimilate and apply it.[29] As the role of firms shifts from internal development and manufacturing to assembling knowledge and components, new capability profiles that allow for efficient and effective scanning of the environment, evaluation of developments outside core areas of activities, and rapid and seamless integration of the external knowledge and components, become fundamental to developing and sustaining competitive advantage. When firms rely heavily on communities, the potential for firms' specific knowledge to provide competitive advantage may be reduced — instead the capability to manage relationships in a way that makes it possible to access, coordinate and use such knowledge becomes critical for competitiveness.

Limitations of the study

While FOSS constitutes a specific, and in some ways extreme, setting, there is a wider need to understand how communities outside the hierarchical control of managers can be used in an effective manner. This is particularly important in settings where firms try to benefit from the online communities that have become such a widespread phenomenon in areas such as the pharmaceutical industry, the production of dictionaries and the manufacture of musical instruments. Although we have outlined the problems encountered, our study did not extend to being able to try to relate these to the ability to be profitable in the long run.

Conclusion

Using communities is a way for firms to increase the total amount of resources they can draw upon in the innovation processes, but there is at the same time a counter-acting need to appropriate the potential value of an innovation by limiting other firms' access to the same resources and information.[30] The distributed nature of the innovation puts additional demands on firms aiming to use the

knowledge residing in communities for their business purposes, and calls for new means to coordinate and control the development and use of knowledge over time. It is apparent that some of these approaches affect the firm's openness: in some cases, firms have to take the deliberate decision to limit it in order to achieve greater management control. In other cases, firms manage the situation by reframing the system boundaries, and complementing traditional means of control and influence with more innovative techniques for activating and involving external knowledge sources. Our study points to an exciting area, where firms are active in creating innovations beyond the four walls of the firm, and are developing strategies to profit from developments in communities they cannot control.

Acknowledgements

We are grateful to the Long Range Planning Editor, Charles Baden-Fuller, and the journal's anonymous reviewers for their superb commentary that has significantly improved the quality of this article. We also thank Ammon Salter and seminar participants at Stanford University for ideas on an earlier manuscript. Dahlander's research was carried out as part of the Innovation and Productivity Grand Challenge with financial support from the Engineering and Physical Sciences Research Council and the Economic and Social Research Council through the AIM initiative. Any errors are our own.

Appendix

Case selection

We made our selection from a list of firms based in the Nordic countries that were working with FOSS as their primary business model, choosing firms that *a priori* appeared to have different business models, thus adopting the theoretical sampling model.[31] Given our theoretical interest in how firms made use of external sources of innovation, we considered it appropriate to have this variation in our case study firms. There is no single database that includes all FOSS firms; it was necessary therefore to screen a large number of firms in order to make our selection. We define a FOSS firm as an independent for-profit organization that uses its expertise in FOSS to develop products or services as its main business. This excludes firms with only some part of their business in FOSS, and includes all those firms that are dedicated FOSS firms. The firms selected for our study are small (<100 employees), but are fairly well known in their respective fields.

Data sources and data analysis

Inductive qualitative methods benefit from some variance in data sources, because this minimizes any biases when examining underlying causes of variation.[32] The data sources fall into three main categories:

- *Firm data*. Secondary sources, including annual reports, company directories, business and specialist press and homepages, were used to gather information on firms. This provided information about their competitive environment, important milestones and the perceptions of outsiders.
- *Interviews with firms*. We conducted 16 semi-structured face-to-face interviews lasting between 0.5–3 hours, which included questions about the company's interaction with communities, the nature of these relations and the rationale for its current strategy. Most interviews were recorded and transcribed. In three cases, respondents did not want their interviews to be taped and extensive and careful notes were taken. Where necessary, the interviews were complemented by telephone conversations and e-mail exchanges. As the firms were all relatively small, we were able to interview most of their managers and founders.
- *The communities.* To obtain more information about the firms' relationships with their respective communities, we followed mailing lists and forums over a three-month period, checking them at least three times a week. In three cases, where a community had been established based on the firms' products, we gathered information about how the firms interacted with users

within the FOSS communities. This comprehensive exercise (which involved screening thousands of e-mail conversations) was designed to provide an idea of how the users in the community interacted, and was an important part of our study.

References

1. H. Chesbrough, *Open Innovation: The New Imperative For Creating and Profiting from Technology*, Harvard Business School Press, Boston (2003); K. Laursen and A. J. Salter, Open for innovation: The role of openness in explaining innovation performance among UK manufacturing firms, *Strategic Management Journal* **27**, 131–150 (2006).
2. E. von Hippel, *The Sources of Innovation*, Oxford University Press, New York (1988).
3. W. W. Powell, K. W. Koput and L. Smith-Doerr, Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology, *Administrative Science Quarterly* **41**(1), 116–145 (1996).
4. H. W. Chesbrough (2003), op. cit. at Ref 1.
5. S. O'Mahony, Guarding the commons: how community managed software projects protect their work, *Research Policy* **32**, 1179–1198 (2003).
6. J. Henkel, Selective revealing in open innovation processes: the case of embedded Linux, *Research Policy* **35**(7), 953–969 (2006).
7. J. Lerner and J. Tirole, Some simple economics of open source, *Journal of Industrial Economics* **52**, 197–234 (2002); E. Raymond, *The Magic Cauldron.* Available from: http://www.catb.org/~esr/writings/magic-cauldron/ (1999).
8. A. Bonaccorsi, C. Rossi and S. Giannangeli, Adaptive entry strategies under dominant standards: hybrid business models in the open source software industry, *Management Science* **52**(7), 1085–1098 (2006); L. Dahlander, Appropriation and appropriability in open source software, *International Journal of Innovation Management* **9**, 259–286 (2005).
9. M. von Zedtwitz and O. Gassmann, Market versus technology drive in R&D internationalization: four different patterns of managing research and development, *Research Policy* **31**, 569–588 (2002).
10. K. Laursen and A. Salter, *My precious technology: the role of legal appropriability strategy in shaping innovative performance*, Working Paper (2006).
11. J. Lerner and J. Tirole (2002), op. cit. at Ref 7.
12. H. Chesbrough and R. S. Rosenbloom, The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies, *Industrial and Corporate Change* **11**(3), 529–555 (2002).
13. R. C. Levin, A. K. Klevorick, R. R. Nelson and S. G. Winter, Appropriating the returns from industrial research and development, *Brookings Papers on Economic Activity* **3**, 783–831 (1987).
14. A. K. Klevorick, R. C. Levin, R. R. Nelson and S. G. Winter, On the sources and significance of interindustry differences in technological opportunities, *Research Policy* **24**, 185–205 (1995); J. P. Liebeskind, Knowledge, strategy and the theory of the firm, *Strategic Management Journal* **17**, 93–107 (1996); D. J. Teece, Profiting from technological innovation: implications for integration, collaboration, licensing and public policy, *Research Policy* **15**, 285–305 (1986).
15. V. Mangematin and M. Callon, Technological competition, strategies of the firms and the choice of the 1st users – the case of road guidance technologies, *Research Policy* **24**(3), 441–458 (1995); M. A. Schilling, Technology success and failure in winner-take-all markets: The impact of learning orientation, timing, and network externalities, *Academy of Management Journal* **45**(2), 387–398 (2002).
16. M. L. Katz and C. Shapiro, Network externalities, competition, and compatibility, *American Economic Review* **75**(3), 424–440 (1985); B. W. Arthur, Competing technologies, increasing returns, and lock-in by historical events, *The Economic Journal* **99**, 116–131 (1989).
17. T. J. Allen, *Managing the Flow of Technology*, MIT Press, Cambridge, MA (1977); W. M. Cohen and D. A. Levinthal, Absorptive capacity: a new perspective on learning and innovation, *Administrative Science Quarterly* **35**(1), 128–152 (1990).
18. S. A. Zahra and G. George, Absorptive capacity: a review, reconceptualisation, and extension, *Academy of Management Review* **27**(2), 185–203 (2002).
19. G. K. Lee and R. E. Cole, 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 (2003).
20. *The Open Source Definition.* Available from: http://www.opensource.org/.

21. S. O'Mahony, The emergence of a new commercial actor: Community managed software projects, unpublished PhD Dissertation, Stanford University (2002); S. O'Mahony, Guarding the commons: how community managed software projects protect their work, *Research Policy* **32**, 1179–1198 (2003).
22. K. Eisenhardt, Building theories from case study research, *Academy of Management Review* **14**(4), 532–550 (1989).
23. M. B. Miles and A. M. Huberman, *Qualitative Data Analysis: a Sourcebook of New Methods*, Sage Publications, California (1984).
24. See http://www.gnu.org/licenses/lgpl.html.
25. J. West, How open is open enough? Melding proprietary and open source platform strategies, *Research Policy* **32**, 1259–1285 (2003).
26. L. Dahlander and M. W. Wallin, A man on the inside: Unlocking communities as complementary assets, *Research Policy* **35**(8), 1243–1259 (2006).
27. H. Tsoukas, The firm as a distributed knowledge system: a constructionist approach, *Strategic Management Journal* **17**, 1–25 (1996).
28. J. S. Brown, Minding and mining the periphery, *Long Range Planning* **37**, 143–151 (2004).
29. R. M. Grant, The resource-based theory of competitive advantage: Implications for strategy formulation, *California Management Review* **33**(3), 114–135 (1991).
30. M. Janssens and C. Steyaert, The world in two and a third way out? The concept of duality in organization theory and practice, *Scandinavian Journal of Management* **15**, 121–139 (1999); M. W. Lewis, Exploring paradox: toward a more comprehensive guide, *Academy of Management Review* **25**(4), 760–776 (2000).
31. B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory*, Aldine, Chicago (1967).
32. B. G. Glaser and A. L. Strauss (1967), op. cit. at Ref 31.

Biographies

Linus Dahlander is a Lecturer and an Advanced Institute of Management Research Fellow at the Tanaka Business School, Imperial College London. His interest is in distributed innovation that transcends organizational boundaries and its implications for individual and firm outcomes. Innovation and Entrepreneurship Group, Tanaka Business School, Imperial College London, South Kensington Campus, London SW7 2AZ UK. Tel: +44 (0)20 594 1955 fax: +44 (0)20 7594 5915 e-mail: l.dahlander@imperial.ac.uk

Mats Magnusson is Director of the Institute for Management of Innovation and Technology in Sweden and Associate Professor at the Department of Technology Management and Economics at Chalmers University of Technology. His main research interests are strategic management and continuous innovation. Center for Business Innovation and RIDE, Department of Technology Management and Economics, Chalmers University of Technology, 412 96 Gothenburg, Sweden e-mail: mats.magnusson@chalmers.se