



HAL
open science

Floss firms, users and communities: a viable match?

Nicolas Jullien, Jean-Benoît Zimmermann

► **To cite this version:**

Nicolas Jullien, Jean-Benoît Zimmermann. Floss firms, users and communities: a viable match?.
Journal of innovation economics, 2011, 1 (7), pp.31-53. 10.3917/jie.007.0031 . halshs-00449535

HAL Id: halshs-00449535

<https://shs.hal.science/halshs-00449535>

Submitted on 22 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GREQAM

**Groupement de Recherche en Economie
Quantitative d'Aix-Marseille - UMR-CNRS 6579
Ecole des Hautes Etudes en Sciences Sociales
Universités d'Aix-Marseille II et III**

**Document de Travail
n°2009-46**

Floss Firms, Users and Communities: A viable Match?

**Nicolas JULLIEN
et Jean-Benoît ZIMMERMANN**

July 2009

DT-GREQAM

FLOSS Firms, Users and Communities: a Viable Match?

Nicolas Jullien, Jean-Benoît Zimmermann

Abstract—The participation of firms in Free/Libre/Open Source Software (FLOSS) communities is growing and is increasingly debated amongst scholars. As [41] explained, FLOSS needs profit and we do not know successful floss products without firms in their ecosystem, either being via the financial support of foundations (Eclipse, Linux) or the commercial offering of products or services based on specific FLOSS products (SQL, RedHat). Various points of view have been proposed, but most of the time, scholars studied either the implication of firms within a community or the integration of floss into their market strategy, but not both. In this article, we plead for a more structured and global analysis, based on industrial economics tools, and thus starting from the basic conditions of the computer market and of the buyers' competence in software development (the 'dominant user's skill'). This conceptual framework helps to distinguish the different roles (understood as 'social roles') firms may play in the FLOSS ecosystem and, specifically the variation in their involvement.

Index Terms—'Free'/'libre' or 'open source' software, Industrial economics, dominant user's skill, firms' roles.

JEL: L11, L15, L22, L86

I. INTRODUCTION

Free/Libre/Open Source Software (FLOSS) has become an important economic issue, and is changing the computer industry, with the decline of some businesses, such as the Unix producers (SCO bought by Caldera, SUN bought by Oracle...) And today, since the IBM's 2001 initiative to invest in Linux other FLOSS products, firms see to integrate the products and the production into their business strategy. And an increasing number of companies are getting involved in the communities of development ([38]), potentially changing the structure and the aim of the volunteer based organization.

The aim of this paper is to propose an analytic framework to better understand the roles played by firms in the communities and to see at which level industrial business strategies can match with the communities' organization [61].

So, if we refer to the levels of analysis of roles proposed by [24], this work tries to complete the studies conducted on FLOSS communities data regarding firms' behavior by first proposing definition of the role of firms based on the market the firms are involved in and second a study of the questions raised by firms' behaviors for the commitment of developers to communities and for the long term stability of the latter. The article starts from a discussion of the literature regarding relationships between firms and Floss, and especially an analysis of the reasons for the firms to get involved in the communities (section 2). We propose thus a classification of

the different businesses built using FLOSS (section 4), based on a segmentation of the computer market, thus of Industrial economics theories (section 3). This allows us to propose a classification of what firms do in FLOSS communities, what we call their "roles" (section 5). In the conclusive section, we discuss the point raised by the study of these roles, both technically (how to study the roles) and strategically (impact of the firms in the involvement of people in the communities and on the stability of the communities).

II. PREVIOUS WORK ON FIRMS AND FLOSS COMMUNITIES.

In the last decade, an abundant and growing literature has discussed this question.

We believe that this literature can be split into two approaches, two points of view based on two data sources, the first looking at the production side and the second at the market side to explain firms' interest in FLOSS.

The production side starts from the communities and evaluates the level and manner of firms' involvement in them. The research agenda, as put forward by [56], is to understand how firms involve themselves in a specific community and what these communities can offer them. From an organization science point of view, the question is how agents organize themselves to manage distributed innovation and under what conditions firms can capture part of this innovation for business purposes. Here, we are close to the analysis of Firms' 'social roles' in Online Communities, social roles understood as [5] defined them. The advantage of such studies is the availability of data from communities.

These roles are not directly studied, as these studies examine the roles played by individual people. Some of them belong to companies and the roles of the companies are deduced from the roles of the users. However, their exploitation leads to crucial results for understanding the links between business and open source. In 2005, [38], analyzing a survey of 287 communities (i.e. people active in FLOSS development projects), showed the importance of business participation in FLOSS communities, as "a majority of [their] respondents are skilled and experienced professionals working in IT-related jobs, with approximately 40 percent being paid to participate in the F/OSS project." [25], studying the "embedded" Linux system, showed that business involvement pursued several strategies and that firms did not reveal all the codes they produced but rather carefully selected their contributions. [15], in a study of the "GNOME" graphic interface project, based on the theory developed by [48], argue that by hiring developers who participate in this development project, firms try to control

a complementary asset important for building their products and services. [28] identify, amongst the various FLOSS projects a “money-driven cluster” where “IT vendors’ motives are economic. In this cluster, significant investments have been made in projects that will serve as complementary assets to drive revenues to vendors’ core businesses”.

However, these authors were looking at already well-established communities, where the software developed is shared by numerous actors, people or firms. This can hardly explain why some companies, like MySQL AB, which owns the entire eponymous database software, open source it and yet remain responsible for most of its development, as if it were the core asset of their business¹.

On the other hand, from a strategy and management sciences point of view, some scholars have looked at the business side and explained the use of FLOSS products in terms of the characteristics of the market and firms’ positioning on this market.

The two main issues here are the definition of a business model (what do you sell when choosing FLOSS?), and the links between a business model and involvement in communities. [7] have surveyed the literature on the different “FLOSS business models” and classified them “into five types: (a) Support Contracts; (b) Split Licensing; (c) Community; (d) Valued-added closed source; (e) Macro R&D Infrastructure”. They have looked at the advantages and disadvantages of each model, proposing a case study for each, but without investigating involvement in the communities, and the market conditions under which each model is most efficient. Surveying Italian firms, [5] have proposed a definition of “FLOSS based business” and different reasons for firms to participate in FLOSS development. But they have not really explained the link between the kind of business model and the degree of involvement in communities.

The first to establish a link between these two aspects were Finnish scholars. Surveying Finnish firms, [13, 14] showed that to a large extent, the variety of firms’ involvement in FLOSS can be understood in the light of the position of software in the firm’s business model (as core activity or not). Again using data on Finnish firms, but at product level, [37] remarked that “it seems that factors other than those typically found to explain differences in entrepreneurial innovation behavior such as firm size and age account for the differences in the product and license type strategies of the software companies. [His] data indicate that the firm ownership structure has a major influence for the software firms’ product-level business strategies” (p. 123).

If these studies prove the link between the market and involvement in FLOSS, they suffer from some limitations. They are rather descriptive and do not propose explanations for the variations in firms’ behaviour in similar markets: why do Asus or Dell install Linux on their computers without

contributing to the development while HP and IBM do? This may be because these authors look at the level of firms without looking at the differences between the different branches of the industry, as [57] did on the server/operating system market (what he called the “platform market”).

The present work therefore belongs to this second category of approaches. It tries to systematize West’s approach to the whole computer industry in order to provide explanations for the variations of firms involvement in FLOSS, in terms of intensity of participations within communities and in terms of responsibility in the organization of the communities.

III. SOME INDUSTRIAL ECONOMICS.

A. Theoretical background.

Industrial economics [44] explains that an industry is characterized by the basic conditions of each kind of activities: characteristics of the products, of the users, hence of the demand, but also of the juridical environment (intellectual property protection, for instance). These basic conditions draw the main aspects of the market structure (source of added value, competitive advantages, sources of barriers to entry) and the nature of the competition (firms behavior, in terms of price, position, etc.) The efficiency of the firms (their performance) would depend on a good adequacy of their strategy (behavior, organization) to the market structure, and to their capacity to reshape this market structure, increasing, for instance, the barriers to entry [51].

According to this, firms FLOSS involvement may be seen as a consequence of the evolution of the structure of the market (new sources of added value or of competitive advantage), but also a consequence of strategic positioning of firms, which may see FLOSS as a tool to improve their efficiency, but also to curve the market in order to attract new customers, or to price new offers.

B. The basic conditions of the computer industry.

There is a wide diversity of actors in the industry in terms of both products and size. Successive waves of innovations and company strategies have led to a progressive reshaping of industry borders and structure. For example, Internet has impacted the software production, pushing firms to integrate more services in their offers, designing new ways of selling software based applications, such as Saas (software as a service) ([12], pp 86-127; [6]).

However, the foundations of the industry (the basic conditions) have remained unchanged, since those described by [23, 60, 46, 12]: IT products are built by assembling hardware and software units in a given architecture, and these products (isolated or integrated into networks) are used as parts of information systems and solutions.

On the basis of such technical organization, it is possible to distinguish three large types of “vertical specialization”: i. component producers, ii. computers and IT devices suppliers, iii. the market for applications. All these segments are concerned with software production, as even chipset manufacturers have to deal with the operating systems embedded

¹This means that any developer/contributor wanting to contribute to the official MySQL product has to transfer her copyright to MySQL. <http://forge.mysql.com/contribute/cla.php>. Once owning the whole copyright, the firm can manage a dual licensing scheme, distributing the product under the license it wants, either GPL or more classical closed license. So a customer that does not wish to reveal further enhancements of the source code has to keep these enhancements to herself.

in the machine integrating their component. But, since the beginning of the 1970s, some firms have specialized in software production and, since the beginning of the 1980s, in software publishing (packaged software). According to [12] (chap. 2), the application market can be split between service and product, and for the product side between business specialized offers and global, platform offers.

These three typical strategic orientations are historically situated and gave rise to a market segmentation that corresponds to different degrees of user sensitivity to price, variety and specificity of the offer.

- 1) The “packaged software” business solution providers: First, during the 1980s and 1990s, a growing number of users sought to reduce the costs and uncertainty related to customized development by acquiring software packages that could be installed and adapted to their specific needs and constraints, with the help of service companies accredited by the software editor ([27], p. 98). These hybrid offers of “packages”, combining standardized goods and customized services, were highly successful in the field of professional dedicated solutions, both for management needs (ERP such as SAP) and for technical software (middleware applications, compilers, development tools like those from Ilog) or branch-specific applications (like software for computer-assisted design or computer-assisted manufacturing from Dassault System or others).

With the development of networks within and, more recently, outside firms, with the spreading of the Internet, a growing number of heterogeneous users have shared a growing number of applications (mainly those allowing information exchange and sharing). This network effect and the related need for standardization have had important consequences on further technological orientation. It was necessary to ensure the availability of a wide portfolio of software tools that could meet the needs of all these, both for expert and for unskilled users, within common or separate organizations. Two kinds of strategies can be observed as responding to this necessity:

- 1) The platform producers: On the one hand, software publishers have broadened the scope of their offer by supplying a variety of application tools that can be combined with their core product or by supplying the latter with multiple versions. This enables them to better meet users’ specific needs while keeping production costs down. The archetypal example of such a “platform strategy” is Microsoft, which now offers different versions of its operating system for servers, corporate users or private individuals, as does its open-source competitor RedHat now. The same kind of strategy can be observed for Oracle, which sells professional applications developed on its database technology, and which has recently bought BEA and SUN, after other takeovers, to enlarge its applications portfolio². Another

²“Ellison saw that if Oracle played its cards right, the confluence of the database, the Internet, and the Web browser could displace the operating system as the focal point of computing and erode Microsoft’s industry dominance”. Brent Schlender, CNN Money, 1999. http://money.cnn.com/magazines/fortune/fortune_archive/1999/05/24/260276/index.htm

illustration is given by Symbian in the field of operating systems (OS) for mobile applications.

- 2) The architects: On the other hand, service companies, especially the large ones, such as IBM and Cap Gemini, have tried to master a wide range of software technologies and products which they can combine and adapt to the constraints and the existing equipment of their customers. They intervene as “architects” of their customers’ information systems. Smaller local services companies have a similar position, also providing infrastructure services but on a smaller scale (single server maintenance instead of a large IT infrastructure) or aimed at specific professional needs (like IT infrastructure maintenance services in the food industry).

All these segments are characterized by strong imperfect competition regimes due to diverse increasing return effects (economies of scale in production, high sunk costs – R&D, distribution channels, etc. -, technological interrelatedness and learning), to use Arthur’s terms [2, 3]. The consequence is that they are dominated by a small number of firms. In these oligopolies, differentiation strategies largely play an important role, and are of two kinds:

- horizontal differentiation either related to the integration of new features and high performance tools or to market segmentation (for instance through hard-soft-content bundling), aiming at segmenting the market between the different participants in the oligopoly;
- vertical differentiation. In that case, firms may propose cheaper products (like notebooks, or Linux based PC servers competing with traditional Unix servers). this is a classic in the computer industry to enter the market (Digital entered the market with a smaller, cheaper computer called a “mini-computer”, as Apple did creating the Personal computer market). They may also improve the offer for the same price (as Microsoft did when adding to its operating system a browser for the same price).

These strategies are efficient if, and only if the users are able to perceive the difference between two offers. Thus the efficiency of these strategies, the possibility of use them depend of users’ skills.

C. The users.

As far as the user side is concerned, we will distinguish three main types of users according to their relation to the product and the technology ([60, 36, 54, 55]). The first is the category of “Naive customers or users” (that we denote N) who are not endowed with noticeable technical skills and do not individually weigh very much in economic terms. They are overall sensitive to prices and even if they may react to new characteristics of products or to branding issues, they are not capable to translate their needs and satisfaction into technical terms. The second is the category of “Kogut-Metiu Users” (KM)³ who are not able to contribute to software development but can generate new features or innovations by revealing their own needs. KM users are sensitive to price and quality

³Our reference to the notion of “frontier-users” put forward by [36].

arguments The third category is that of the “Von Hippel Users” (VH) who may act as “sources of innovation” ([54, 53]) able to contribute to software development by proposing improvements or modifications, developing it by themselves or at least able to design the technical specifications.

Users play a double role, deriving from both their economic and technical standing. Depending on the market, and especially their bargaining power in it, the users are more or less able to select the (technical) offers. At one extreme, users and contracts in the global service/architects market are related to large structures, with substantial buying capacities and generally endowed with significant technical skills. So they are likely to influence economic and technical choices. At the other extreme, low price computers address a mass market where individual users, in their vast majority have little budget and/or few skills. Their influence on market evolution is negligible at an individual level but of global importance in terms of elasticity to prices. But this analysis should be nuanced in the case of intermediation by a “prescriber”, who orders and defines the characteristics for a large number of machines, destined for mass distribution by his/her own means (local government for secondary schools, education in rural area in developing countries⁴...) That’s the reason why, when speaking about the “user”, we mean the person who negotiates or chooses the characteristics of the good, who is not always the end user. Of course different types of users co-exist in any given market. But the dispersion of users’ skills in the related technology and more particularly in software doesn’t follow the same distribution from one segment to another. Even if skilled users are likely to be found in any market, they may represent a share too small to play a significant role in it and catch the interest of the firms concerned for their specific demand. Conversely, thanks to the Internet, and regarding FLOSS, a handful of very talented users around the world can weigh enough together to develop a FLOSS alternative to private offers and contribute to the emergence of a FLOSS business offer. So, what we denote users’ skills appears as a subtle mix between competences and number, which could yield a “weighted sum of competences” [32].

D. Structure and strategies.

So, in a market where the dominant user is naive, firms can only differentiate on prices, as it is the only signal understood. The more competent users are, the more firms can differentiate vertically on quality or horizontally via the creation of niche products for specific needs and users.

If we focus on computer and software and services, we see that users’ competency is a key parameter for the structure of the industry and for explaining how firms consider software and invest in development.

1) *Hardware.*: When speaking of computers, we think about machines which are more or less dedicated to specific uses. At one extreme computers can be used for a wide scope of applications provided by the software that is acquired

and installed on them. At the other extreme, video game consoles or multimedia players are devoted to a single range of applications, while in between, mobile devices like PDAs or mobile phones are built to support a growing number of applications⁵.

Vertical competitive advantage is given by better performances/cost ratios (for instance lower price laptops or better computation capacities for servers or high quality laptops), while horizontal differentiation is based on the integration of new features and high performance tools (as Samsung did when proposing a fold display mobile phone) or to market segmentation through hard-soft-content bundling on new features or applications (Mario Broth video games being only on Nintendo machines).

But regarding the objectives of these machines, and thus the skill of the people buying them, the structure of the markets and of the competition varies. We will take the example of the computer market to illustrate this.

Servers are intended to manage, deliver and protect information on the networks. They must be high-performance, stable, but also compatible with network standards and are bought by VH users. Microcomputers (with a growing market share for laptop computers) are used by end users, mainly as personal computers. In the server market, several Unix systems still exist, and this is case of horizontal differentiation as they are not compatible, so users have to choose between them. For high end customers or needs, mainframes still exist with dedicated operating systems. If we look at FLOSS Unix-like systems, some users prefer BSD (being being FreeBSD, OpenBSD or NetBSD) to Linux. So, even if a growing part of the market is answered by PC servers running either Linux or Microsoft, it is clear that the quality purpose and niche market strategies are possible, because users are able to evaluate the performance and the matching of the offers to their needs, and are ready to pay for that. In the Personal computer market, Apple has a marginal market share, as does Linux, and the Windows-Intel couple dominates the market. IBM sold its PC division to Lenovo in 2004, because it was not profitable any more, after Compaq dropped in the mid 1990s and the difficulties of Dell today⁶ proves that the PC market is dominated by a price war. This is not surprising as the dominant user is naive and thus only price sensitive. The consequence is that firms are continuously seeking to decrease their cost and their price, as it is difficult for them to differentiate horizontally.

2) *Software and service.*: Platform manufacturers may have been the most studied. In a nutshell, they are involved in a classic arbitration in the dispute over standards⁷: to attract the maximum of users on the platform to attract the maximum of applications producers, and vice et versa. Linux distribution editors story is another example of the

⁵This distinction between specialized and generalist devices is evolving, as Sony intends its PS3 to be the media center at home. But this has not so far impacted on the industrial structure.

⁶http://economictimes.indiatimes.com/Infotech/Hardware/Dell_may_sell_its_plants_worldwide_Reports/articleshow/3449300.cms

⁷On standard theories, see the discussions by [33, 34, 35], [48], [40, 39], and for a review of literature, [57, 58].

⁴See, for instance, the competition between Microsoft and Mandriva to supply 17,000 computers in Nigeria. <http://www.computerworlduk.com/management/government-law/public-sector/news/index.cfm?newsid=6124>

importance of the competence of the demand to create a market. RedHat, SuSE, Mandriva (formally Mandrakesoft), were among the first commercial actors to enter the market using FLOSS. This could be seen as obvious on a mass market with rather naive users and a significant prices based competition. But today, the retail store sales of OS package represent a negligible part of the revenue of such firms⁸, and a major part targets the industrial market. One can explain this fact by the development of broadband connexion. But more than that, the competences of the users matter. PCs are shipped with an already installed OS and few of the buyers are skilled enough to install a different OS. Additionally they have incentives to do it because the first installed OS has already been paid for with the computer. On the emerging OS for PC server market things work differently. Most of the users, of VH or KM type, are aware of the technical issues for installing and configuring an OS. It is also easier to buy a machine without an operating system installed. FLOSS gives them access to a more open and more adaptable Unix-like operating system, than they could find in the traditional Unix offer, and they are able to choose the Unix they prefer. So, even if they are less price sensitive, FLOSS based servers may help to differentiate vertically (better quality over price ratio) and horizontally (with the existence of niche Unix).

The practice of combined offers, or packages, integrating a standard base and customized services has made its mark in the professional solutions, whether it is for the company management systems (ERP, whose symbolic model is SAP), or the "IT" tools ("middleware" applications, compilers, development tools such as those proposed by the Ilog company), or the solutions specific to a branch, a profession (such as the subsequent version of computer-aided design proposed by the company Dassault Systems). The producer sells "three A services" [30]: quality Assurance, Adaptation (more or less rapidly) to the user's needs, and user Assistance to use the tool. This is the model of "sustained technical capacity" [17, 22]. The core competence of the firm is in this capacity to make the product evolve to follow the needs of the users, but to make this evolution "sustainable" (ie making it still appropriable and without a decrease in performance, for bug reasons). If these tools are professional tools, users are skilled enough to express their needs requirement (for instance, if they are doctors, that the product is up to date regarding drugs and drugs interaction). But they are not always skilled enough in computer science to develop these requirements by themselves, or even to translate them into tender specifications or in technical specifications. Here, it is the content of the users' feedback which may vary according to their computer skills.

Service companies, or "IT Architects", and especially the largest (IBM, Cap Gemini), endeavor to develop a global ap-

proach to IS and company organization (by acquiring strategic consultancy companies such as Ernst & Young), while remaining less dependent on one type of software, in order to be able to adapt to the constraints and to the current circumstances of these customers. But the retail service companies behave in the same way, supplying infrastructure on a more local and smaller scale (maintenance of a single server, instead of a global infrastructure), either at a more specialized level, for example in terms of sector (e.g. maintenance services for the food-processing industry), or on a more reduced software base (distributors-installers-adapters of one of the platforms, these are Microsoft, Oracle, or RedHat "certified" companies). The vocation of all these companies is to develop, in the customer's interest, individualized solutions and to support these solutions. We are approaching what [17, 22] described as the "provision of human capacities", in the sense that what makes their singularity (or their core competence) is that they bring together a team of specialists of differing software, but also of customers' vocational specialists. In the following, we will call this "architect strategy". In other words, the efficiency of these firms is in producing tender specification or technical specifications meeting their clients' needs. If these companies are rather technical agnostics as they have to install the tools their clients need (or want), it is obvious that the more they dominate a tool, the easier its adaptation is, the easier their job is. This point enlarges the strategy field, as firms may differentiate vertically (increasing the number of tools mastered, the number of professional domains covered), but also horizontally, specializing on a domain or a software, as the SAP consultants. But, once again, in any case, the more computer skilled their clients are, the easier the discussion may be [16].

3) *A summary of firms' strategies regarding software.*
This overview of the computer market may be summarized in the following way: the more skilled people are, the more the strategies increase because it is easier to differentiate horizontally, trying to address users' needs ever closer, thus creating niche markets. When users are too computer illiterate, competition is restricted to price. This explains the industrial appropriation of FLOSS, and its variance from one market to another and the competitive advantage FLOSS gives.

IV. FLOSS AS A STRATEGIC TOOL IN FIRMS' HANDS.

It is clear that the cost of licenses offers a price advantage. Moreover, the fact that the customer can evaluate the product without buying a license is also an advantage in terms of dissemination. But this is not different from a freeware, and does not explain the success of FLOSS. Actually, we will show that FLOSS (and FLOSS based business) succeeds when this success is based on the open source and community feedback framework assets.

A. Hardware.

This may be the sector where the link between structure of the market and FLOSS business practices is the clearest.

⁸RedHat stopped this activity (see financial report 2006, p. 31); the consumer market (including distributors, OEM sales, e-commerce and Club) represented 2.54Md (45% of the total earnings) showing a 23.4% decrease for Mandriva 2005-2006 fiscal year; SuSE has been bought by Novell, so these revenues are diluted.

In the server market, producers have habitually provided proprietary solutions with proprietary Unix⁹. The rise of PC servers has permitted some users to avoid such a bundling problem; moreover, using Linux or another free Unix means a cheaper offer (vertical advantage) reusing Unix programs (content) portfolio. Here suppliers are dealing with highly-skilled VH clients that have forced them to adopt FLOSS. The competitive advantage of FLOSS is its openness, allowing it to be tuned to specific needs, as proven by Google, which runs more than 40,000 servers with “customized versions of Linux”¹⁰.

On the segment of notebooks, where users are mainly naive, competition is overall based on prices. When Asus entered the market with its eee-PC, it used Linux for price reasons, because Microsoft Windows Vista was too costly in terms of resources needed and price to be competitive. Since then, considering the success of this market, Microsoft has designed a specific, downgraded version of Windows XP for these computers, and today, Linux market share has dropped to around 10%¹¹. In that case free software is considered as a freeware and does not seem to provide any competitive advantage. It has only been used to force the monopoly to drop its prices.

B. Software and service.

1) *Software platforms.*: We already shown that Linux has succeeded where users were VH. Of course for price reasons, but also for quality reasons: people could choose a system better adapted to their needs as it is more adaptable, as did Google. For a platform editor, the attractiveness of its platform is a growing function of the application available on it. The asset of a FLOSS platform is to allow the creation of a community of application providers, which will adapt their product to their platform. And, as these applications are not at the same level of maturity, since they may be incompatible, there is a need for an actor granting this compatibility, helping firms to select the applications they need. In a word, the “3A services” (assistance, assurance and adaptation to the use).

RedHat is the best example of such a strategy, with on one hand, the Fedora community to foster “innovation”¹², the availability of applications on a Linux platform, and on the other hand, RedHat enterprise, when you want to buy assurance and assistance¹³.

2) *Packages.*: We said that a growing source of revenue comes from, again, “3A services”. Currently, the main evolution for those firms is to switch from a demand pull strategy (functionalities are developed to stimulate/create the demand) to an ‘on-demand’ development (development when required and paid for or carried out by the users). Therefore, FLOSS is used to increase the business feedback from users and by considering openness as a way to reduce transaction costs and

as a signal of quality. This explains why open source business products are developed mainly in “business” software (ERP, computer infrastructure software like compilers), where users ready to pay for configuration, maintenance or assistance services are numerous. We will distinguish two kind of structures, again regarding the computer skills of the users:

- 1) When users are VH software professionals, we find one firm organizing cooperation around its brand named product. The producer approves the contributions, ensures stability of the tool and helps developers to use it. If some individual contributor becomes important (in terms of contribution volume/quality/innovative aspect), s/he may be hired by a producer, with reduced recruitment costs and risks (ACT or MySQL but also some small services companies are using this method). By contributing to innovation, the developers (and possibly companies using the tool), are therefore guaranteed that their needs will be taken into account more quickly and integrated into the product (which is a fundamental factor in reducing costs, according to [54]).
- 2) When users are more KM, firms are more service oriented. The open source asset is more in being a flexible, adaptable input, developed by a consortium of information system consultants sharing and co-developing the tools they are basing their business on, with sometimes a two-level organization, where a software producer and editor of a tool deals with information system resellers. The text-book examples are Compiere ERP and CRM, or Zope CMS, where there is an editing firm, which sells its services and products as in case one, but mainly to “partners”, service companies, as shown in figure 1 on the following page. In the second case, resellers act as service, architect companies we will discuss in the following paragraph. In both cases, the interest of FLOSS is its flexibility which is used to adapt itself to the client’s needs and the license, which guarantees that the product will always be available, as the modifications performed by other firms. the consortium organization, decreases the cost of development for each member (as it is shared) and facilitates and accelerates the achievement of a global presence for the product.

When naive users are dominant, it does not seem that firms manage to do a direct business based on FLOSS products. Even if some FLOSS offers exist for that segment, such as Open Office or Firefox, their market share remains small¹⁴. We do not know firms dedicated to them, and, according to us, firms’ support is more a consequence of platform or hardware providers strategies: these actors need these commodities for their platform to be adopted by VH or KM users, and FLOSS is a means to create a consortium to develop it (as SUN does by supporting Open Office development).

⁹See [57] for a full discussion of FLOSS strategies in that sector .

¹⁰See http://en.wikipedia.org/wiki/Google_platform for a summary of the Google platform and usefull links on that topic and [47] for a whole presentation of Google.

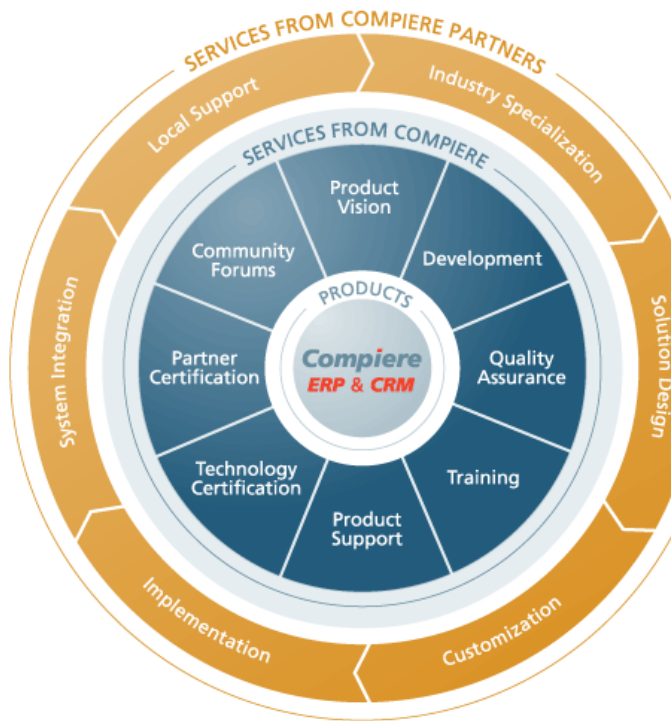
¹¹<http://www.i4u.com/article23707.html>

¹²See the presentation of Fedora at:<http://fedoraproject.org/wiki/Overview>

¹³Also very well explained on RedHat’s Web: <http://www.redhat.com/rhel/>

¹⁴Some sites estimate Firefox market share around 20% (see <http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0>) but they probably over estimate it, as they look at the browser used to visit site, which thus favor the browsers used by Web intensive users, which are more skilled than the mean user, and may use more FLOSS.

Figure 1: The Compiere “ecosystem” (taken from: <http://www.compiere.com/partners/>)



3) *Architects.*: As Horn (2004) points out, assembling components requires access to the source codes (problem of compatibility), and their adaptation to different needs (of users and other components). They must be available in the form of open-source software (therefore legally modifiable). The competitive advantage in using FLOSS, in addition to price, is therefore the ability to offer an assembled set of components with greater interoperability, which should increase the quality of the final product, on a market where the quality of services is one of the recurrent problems (see De Bandt, 1995). Revenues are generated by assembling and adaptation services, as is the case for any traditional service company. The only uncertainty about the model concerns the availability of the components: who will develop them and who will maintain them? Moreover, the customers of these companies may already have (proprietary) programs installed that need to be taken into account. In the end, an open source strategy could even be a guarantee of means (maximum use of free software), but not a guarantee of the results (use of only free software), unless the customer requests this, since in this situation, s/he has the last word.

Two kinds of firms use FLOSS today: newcomers who specialize in FLOSS architecture, using FLOSS as a vertical (price) and horizontal differentiation asset¹⁵, and incumbents, such as IBM for its service activities. Traditional service firms like Cap Gemini are more agnostic with regard to the technologies

¹⁵As explained by [45], one of the main strategy for newcomers in technological market is technological differentiation. Basing its offer on new FLOSS products can be seen as a way for new service companies to differentiate.

used and the intellectual property regime involved. They will generally follow the customers’ demand which depends on their ability to keep up with the development of the project. These customers are most often large organizations, skilled computer users that are receptive to the opportunity to integrate the most advanced software components, developed under FLOSS licenses. So they are becoming increasingly involved in FLOSS as the market grows and matures¹⁶. Global service firms’ Web site are quite explicit on this strategy¹⁷.

C. The dependence of users’ skill.

Table I summarizes the section.

What seems clear after this rather qualitative analysis, and was formally demonstrated by [32], it that the skill of the users matters for understanding the level of implication of firms in FLOSS. When users are naive, firms may use FLOSS, but only for price reasons, in the same way as they could use freeware. The more VH users are, the more complex strategies regarding FLOSS are, and the greater firms’ involvement and participation. In some cases, when users are VH, firms may even produce FLOSS and animate the community like Ada Core Technology for Ada 2005 and MySQL AB for MySQL data bases do. But in any case, FLOSS is regarded as open source software. This means that in that case, firms use FLOSS for technical reasons (sustainability, flexibility) and for innovative reasons (increasing the speed and quality of feedback).

V. THE ROLE OF FIRMS IN FLOSS COMMUNITIES.

The consequence of firms market position is that they will behave differently according to the significance of FLOSS software or FLOSS community in the specificity of its offer: is it a complementary or a core asset for this offer? According to the theory [48], if yes, firm should invest a lot to manage this asset, if not, it should not invest at all, buying or using it as it is (as a component “of the shelves”). This has been confirmed by a survey of FLOSS business based firms [31].

A. Naive users, the freeware strategy, the community as a commodity.

As, in that case, the aim of using FLOSS is to propose the lowest price possible, firms will not invest in FLOSS development more than the effort needed to adapt the software to their product(s): this would increase their cost. So FLOSS may be seen as a free commodity, a freeware.

Of course, one might put forward that such strategies have a flavor of free-riding and the risk is to demotivate the most committed people in the communities, seeing others drawing

¹⁶In 2005, Gartner forecasted that “[in] 2008, 95 percent of Global 2000 organizations will have formal open-source acquisition and management strategies” (<http://www.gartner.com/it/page.jsp?id=492152>). In their 2008 study, they said that “Adoption of open-source software (OSS) is becoming pervasive, with 85 percent of companies surveyed currently using OSS in their enterprises and the remaining 15 percent expecting to in the next 12 month” (<http://www.gartner.com/it/page.jsp?id=801412>).

¹⁷See Capgemini’s for instance: http://www.us.capgemini.com/services/serv_overview.asp?ServID=30

Table I: Firms' FLOSS strategy according to the skill of the users.

	Naive	Kogut-Metiu	von Hippel
Hardware	FLOSS as a Freeware (Linux, with Open Office and Firefox). Price differentiation. Ex.: Asus' EeePC.	FLOSS to create a consortium on commodities development. No business. Ex.: Open Office and Firefox are installed in computer to increase the applications available for the same price.	FLOSS as technical tool increasing the quality over price ratio via horizontal (adaptation to the needs) and vertical (cheaper use) differentiation. Ex.: PC servers running Linux or *BSD.
Platforms	No FLOSS strategy	FLOSS to create a consortium on commodities development. No business. Ex.: Open Office and Firefox.	FLOSS to foster applications availability. Firms sell selection, compatibility setting and stability over time. Ex.: RedHat
Business software	No FLOSS strategy	FLOSS to create a consortium of consulting firms selling adaptation of the product.	FLOSS to create a closer relationship with users (quicker and better feedbacks). Firms sell 3A services. Ex.: MySQL or Ada Core Technology.
Service	FLOSS to propose cheaper services? No example known.	FLOSS as a flexible commodity to propose/sell cheaper service solutions. Ex.: Compiere ecosystem.	FLOSS as a flexible commodity to meet users' demand and to propose more flexible solutions. Ex.: Cap Gemini, IBM global Service

profits from their own voluntary work¹⁸. However, by adopting FLOSS products, those firms participate in the expansion of the FLOSS users' network for given open software, such as the operating system Linux or the "Open Office" office application suite. In a competition regime in which the battle for network externalities and standards play a crucial role, this may be considered as a strong boost to FLOSS, that can at least win the community's neutrality if not approval.

B. Business packages for VH users, the community, a core asset.

As far as business packages are concerned, the specific asset of the producer lies in its package knowledge and in its capacity to manage the dynamics of evolution. This makes the open sourcing of a software the specific asset of the firm which owns it: on the technology markets where the customers are computing developers, revealing the code facilitates cooperation. The producer organizes the collaboration in a "symbiotic" relationship (using the terms of [13]). Developers (possibly companies using the tool), by providing their own innovations, are thereby assured that their needs will be taken into account more rapidly and integrated into the product, a crucial point to reduce their costs ([54]); from the producer's point of view, this decrease the R&D cost as the users provide him/her with new feature requirements and, more original, implementation; on the other hand, only the one who integrates contributions is capable of verifying and of guaranteeing their correct functioning and to help clients to use it. So, a FLOSS based package model means that the firms which publish the software remain heavily involved in its development in order to control it. As their core competence lies on the management of the software edited, the companies should only invest in the software they edit, and the involvement of salaried developers in other projects should not be encouraged¹⁹.

C. Services to VH and KM users, the community as an complementary asset.

In between these two extreme cases of involvement in communities are service firms. The only uncertainty in their

FLOSS business model arises from the availability and the quality of FLOSS components: who develop(s) them, who maintain(s) them? So they need to evaluate these components and to monitor their evolution²⁰. This need for evaluation and control increasingly depends on the importance of the component for their business and that of their clients. We may even formulate the hypothesis that the more skilled the users are, the more the firm must master the technology, because of the growing level of complexity of the feedbacks and demand.

And to be able to integrate knowledge and innovation from the open-source communities, open-source firms have to develop internally efficient capabilities of absorption, an essential condition to capitalize and internalize the communities' contribution and the users' feedbacks to improve their own product quality. [14] working on the relations between firms and open-source communities show that those firms need "to develop sufficient absorptive capacity to benefit from external developments, not only to identify useful external knowledge, but also to assimilate and apply it". This is what has been called a "commensalistic approach" [13]. This corresponds to the more general assertion from [9, 10] about the necessity for a firm to make internal efforts of R&D a prerequisite for the absorption of external technology.

We consider this reflects a change in the technologies used, thus of the complementary assets these firms need to manage, not really in the core competences. Traditional architect firms are not involve in FLOSS development, as they do not use these technologies. But they may have other processes for monitoring the evolution of the complementary asset, the technologies they use. They may participate in editors' training sessions, or conclude "global alliance" with their key partners, as Cap Gemini does²¹.

²⁰Considering the evaluation part, it is worth noting that the main service companies in France have published tools to evaluate FLOSS, and they use this as a commercial argument (we would say a signal) to their clients on their capacity of evaluating these products. See, for Atos, http://www.uk.atosorigin.com/en-uk/services/solutions/systems_integration/technologies_expertise/open_source/default.htm, and for Cap Gemini, <http://www.capgemini.com/services/technology-services/open-source/solutions/>.

²¹<http://www.capgemini.com/collaboration/alliancepartners/> for classical alliance, and http://searchsystemschannel.techtarget.com/news/article/0,289142,sid99_gci1261207,00.html with those done with open source world.

¹⁸See [21] for a discussion of that particular point.

¹⁹ [31] verified these two points.

D. Roles of Firms in FLOSS communities.

These results are consistent with Teece’s theory [48, 49], if considering the FLOSS community and the evolution of a technology as the asset (see table II). If firms see it as a source of technology they do not invest. They do invest a lot if it is the core of their business to guarantee a efficient integration of the innovations coming from the community to this community. In between these two extrem cases, the community is seen as a complementary asset, as firms try to participate to follow and control the innovations coming from the users.

Thus firms may undertake the different “roles” of the FLOSS organization onion model²² (figure 2): core developer (like MySQL), developer (IBM with Apache), bugs fixers or reporter (Compiere “partner” companies) or user (Asus), and this mainly depends on their users.

This raises a series of questions on the relationship between firms and communities which we will discuss in the following conclusive section.

VI. CONCLUSION. TRACKING AND ASSESSING THE ROLES OF FIRMS IN FLOSS COMMUNITIES.

Investigating the role of firms in communities, and the matching with the theory we have proposed appeals for studies, in, at least, two directions, where precursor works exist, but which are still mainly unexplored.

The first may appear technical, as it deals with data collection: how can we measure firms’ role in the communities, where the participation is individual based?

The second deals with the interaction between two kind of organizations: firm and community. In terms of FLOSS model efficiency, what competitive advantage firm participation gives, and what that means for firm organization (the management of their employees involved in the community) and for community organization (in terms of governance).

Actually, there is a third direction directly connected with the two first but a bit peripheral in the focus of this article: the impact of a commercial involvement on FLOSS on people’s involvement²³.

A. Measuring the role of Firms.

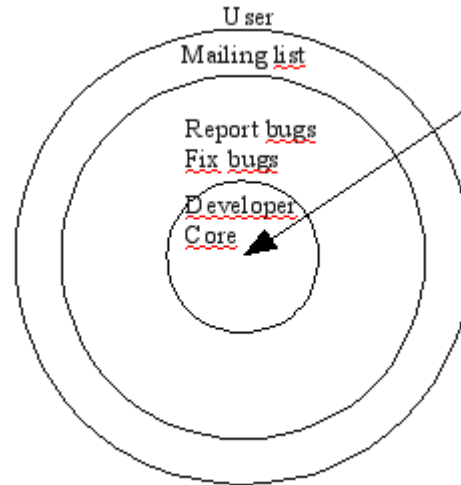
The main issue in this part is that organizations do not contribute by themselves to the projects, but via people hired

²²This model is described by [26] and has been proposed by [11]. In a career in a community, the successive tasks a developer may do are to use the program, to use the mailing list(s), to report bugs and to fix them, to be a core developer.

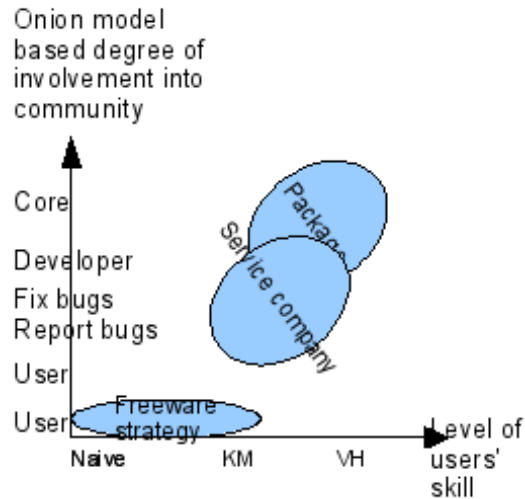
²³The study of FLOSS participants’ motivations has already generated a vast literature, reviewed by [38] and [43], [42] among others. The role of a person evolves with time [29], as evolve the reasons why s/he participates [19]. May it be called the “process of joining” [26], the “role migration” [29] or the “co-evolution of systems and communities” [59], the evolution follows a “canonical path”, the “onion model” [26] from none to center responsibility for the future of the community. This is closed to the sociological notion of “career” [4], as pointed out by [19] [43].

With firms, the career is becoming twofold: the community participant’s one is connected with a more traditional one, the professional career [52]. This has potential consequences on the design of the canonical path, but also on the motivations of “free” developers.

Figure 2: Onion model based level of involvement in FLOSS communities targeted by firms regarding their business and the skill of their users.



(a) Onion model of the level of involvement in FLOSS communities (from [26])



(b) Level of users’ skill and level of involvement of firms in FLOSS communities.

to do so. As already said, some works exist on that topic (see [15]) and they start from firms’ people’s behavior in a community, so using the data produced by the community. This suffers from some limitations.

1) *Collecting the behaviors of firms.*: The way scholars collect data on that topic is to look at the data produced by

To have some responsibilities in a community, you have to “participate”, to comit to this community, and to behave as the community expects you to behave. Firms change this toward more strategic involvement: one can be hired to participate to a community, but with a level of involvement bounded by the employer (ex of service firm). Can you still talk of “freedom” (of expression, of choice, of contribution), which is “one of the roots of the FLOSS culture” [42] and a well known incentive to participate, when people are paid to make in practice employer’s strategy of participation?

FLOSS employees may face a contradiction, a “tension” [50] between the commitment to the community and to their employer.

Table II: Firms' investment in the asset "community".

	single firm community	consortium	open community
package firm	one dominant	contribute marginally if needed (to make its package work correctly)	contribute marginally if needed (to make its package work correctly)
service firm	from technology taker to marginal contribution in function of the importance of the technology	from technology taker to marginal contribution in function of the importance of the technology	from technology taker to marginal contribution in function of the importance of the technology
freeware strategy	technology taker	technology taker	technology taker

the communities (email lists, source code produced and signed, etc.) People are said belonging to a company when they sign with a company address.

The main limitation of this regards the exhaustivity and the coherence of the data collected: some developers do not contribute using their employer's address, and others do but without employer's agreement. However, when firms commit themselves to a community (as IBM could have done to Linux one), this technique may be usable to track their involvement and their evolution in time.

But as [24] pointed out for people, looking at what firms do does not provide information on why they do it, or if this is driven by the management, in a word, on firms' motivations.

2) *Surveying firms' strategies.*: Looking, in the same time, at firms' business may provide a good validation of the firms' role theory we developed here: the link between their involvement and their market.

So, we plead for completing studies of firms' behaviors by questionnaires, or interviews, as [43] could have done to track people's motivations to participate. According to us, a point that should particularly be studied is the missions given by the firms to developers participating to a community: are they hired to participate, are they hired because they participate, and do they have a specific goal (to reach a certain level, to take in charge certain part of the project), or participation is the only thing required?

This is important to understand how much the employees will involve, but also what firms bring to the community and thus to the FLOSS ecosystem in a whole.

B. Integrating firms into the studies of FLOSS ecosystem.

The origin of the open-source rationale remains that of developer-users pooling their development efforts for their own needs, aiming at better access to efficient tools for everyone. Either this volunteer collaboration is initiated by individuals or by a firm, a key condition of the success is users-developers adhesion.

The management of the relation between an open-source firm and the related communities is thus of crucial importance, as a recent literature has already emphasized. [1] have observed that to preserve the co-existence and co-operation of two types of organisation that are based on remote albeit not contradictory rationales, firms must, in a nutshell, "not seek to dominate and control process", "provide professional management and business expertise" and "help establish an open and trusted ecosystem". They view such interaction as rather osmotic than parasitic, as firm's resources reinforce

communities sustainability. These considerations could be easily extended from the sole framework of the communities to the whole users base.

In the other hand, regarding community governance, do organizations weight the same as actors like individual benevolent, or, by their size, their financial capacities, do they curve the evolution of the communities (in terms of organization, goals, etc.)? For instance, what does "IBM is treated the same as any other member of the Apache Group, albeit one with deeper pockets and more developer resources to spare" ([20], p. 2) mean in concrete?

Actually, if we look at the level of implication of firms, summarized in table II, firms' involvement can be related to different structure of firm community relation, or kind of structuration of communities by firms:

- single firm community, when a firm control the kernel and thus make the main decisions for the evolution of the software and must animate the community of its users.
- consortium, when firms dominate and cooperate to develop a platform (Apache, Linux). More often in that case there exist a foundation where the main decisions are taken, and which can hire the main responsables for the project.
- open communities, where firms do not seem to play a dominant role.

In the two first cases, FLOSS communities are more firms communities, regarding FLOSS as a way to organize the cooperation. So the risk is that individual users are of less weight and thus the meritocracy may be less and less true. This may decrease the level of new demands, new propositions provided to the community, and thus its innovative rate. This is espacialy true for the consortium communities, where the actor may focus on the negociation of a common standard and do not try to integrate as much innovation as possible.

In the third case, firms have no real power, but can be more disturbing for the stability of the community: if they use without contributing, they may discourage the main contributors ([21]). If the dynamic of the community decreases, firms will not invest to maintain the community and may forbid their employees to invest themselves more.

C. Lessons for open innovation regimes.

The FLOSS movement has sometimes been presented as a canonical model of production for the open innovation paradigm [8], and even for the knowledge society. If so, open development may develop in fields where users are skilled enough to initiate the development of open knowledge and

have enough market power to force the traditional producers to shift to an open model. The major risk in that model is to kill the goose that lays the golden eggs discouraging individual participation by over control or non cooperative behaviors.

These conditions being respected, the open IP regime can be seen as a very efficient solution to the Schumpeterian dilemma in so far as it permits a wide diffusion of knowledge, while encouraging innovation, as producers are incited to contribute to the development of the product they use/sell.

This regime could be named “VH open innovation regime”, in reference to Von Hippel’s seminal work on users as innovators [54]. Open initiatives have been launched in many industries, such as biotech, remote sensing and chip design. Most of the time, their chances of success are evaluated in terms of the motivation of the participants and the stability of the “community”. Our contribution argues for more economic aspects.

REFERENCES

- [1] P. J. Ågerfalk and B. T. Fitzgerald. Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. *MIS Quarterly*, 32(2):385–409, 2007.
- [2] W. B. Arthur. Competing technologies, increasing returns and lock-in by historical events: The dynamics of allocations under increasing returns to scale. *Economic Journal*, 99:116–131, 1989.
- [3] W. B. Arthur. *Increasing returns and Path dependence in the Economy*. University of Michigan Press, 1994.
- [4] H. S. Becker. *Outsiders. Studies in the Sociology of Deviance*. Free Press, 1963.
- [5] A. Bonaccorsi, S. Giannangeli, and Rossi C. Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science Volume*, 52(7):1085–1098, July 2006.
- [6] Martin Campbell-Kelly and Daniel D. Garcia-Swartz. From Products to Services: The Software Industry in the Internet Era. *SSRN eLibrary*, 2007.
- [7] V. Chang, H. Mills, and S. Newhouse. From open source to long-term sustainability: Review of business models and case studies. In *All Hands Meeting 2007, OMII-UK Workshop*, Nottigham, UK, 10 September - 13 September 2007.
- [8] H. Chesbrough. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press, Boston, MA, 2003.
- [9] W. M. Cohen and D. A. Levinthal. Innovation and learning: The two faces of r&d. *Economic Journal*, 99:569–596, 1989.
- [10] W. M. Cohen and D. A. Levinthal. Absorbitive capacity, a new perspective of learning and innovation. *Administrative Science Quarterly*, 35:128–152, 1990.
- [11] K. Crowston and J. Howison. The social structure of open source software development teams. International Conference on Information Systems, 2003.
- [12] M. Cusumano. *The Business of software: what every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad*. Free Press, New York, 2004.
- [13] L. Dahlander and M. G. Magnusson. Relationships between open source software companies and communities: Observations from nordic firms. *Research Policy*, 34:481–493, 2005.
- [14] L. Dahlander and M. G. Magnusson. How do firms make use of open source communities? *Long Range Planning*, 41 (2008):629–649, 2008.
- [15] L. Dahlander and M. W. Wallin. A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35:1243–1259, 2006.
- [16] J. De Bandt. Les marchés de services informationnels: quelles garanties pour le client, consommateur ou partenaire? *Revue d’économie industrielle*, 86, 4^e trimestre:61–84, 1998.
- [17] J.-C. Delaunay and J. Gadray. *Services in Economic Thought: Three Centuries of Debate*. Kluwer, Dordrecht, 1992.
- [18] D. Demazière, F. Horn, and N. Jullien. Le travail des développeurs de logiciels libres. la mobilisation dans des "communautés distantes". *Cahiers lillois d’économie et de sociologie*, (46):171–194, June 2005. 2e semestre.
- [19] D. Demazière, F. Horn, and N. Jullien. How free software developers work: The mobilization of ‘distant communities, 2006. A former version (in French) has been published in [18].
- [20] Roy T. Fielding. Shared leadership in the apache project. *Commun. ACM*, 42(4):42–43, 1999.
- [21] D. Foray, S. Thoron, and J.-B. Zimmermann. *Open Software: Knowledge Openness and Cooperation in Cyberspace*. 2007.
- [22] J. Gadray. *L’économie des services*. coll. Repères, La Découverte, Paris, 1996. réédition de 1992.
- [23] L.-A. Gérard-Varet and J.-B. Zimmermann. Concept de produit informatique et comportement des agents de l’industrie. In *colloque "Structures économiques et économétrie"*, May 1985.
- [24] Eric Gleave, Howard T. Welser, thomas M. Lento, and Marc A. Smith. A conceptual and operational definition of ‘social role’ in online community. 42nd Hawaii International Conference on System Sciences, IEEE, January 2009.
- [25] J. Henkel. The jukebox mode of innovation - a model of commercial open source development. *DRUID Working Paper*, (06-25), 2006.
- [26] Israel Herraiz, Gregorio Robles, Juan José Amor, Teófilo Romera, and Jesús M. González Barahona. The processes of joining in global distributed software projects. In *GSD ’06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 27–33, New York, NY, USA, 2006. ACM.
- [27] François Horn. *L’économie des logiciels*. repères, la Découverte, 2004.
- [28] M. Iansiti and G. L. Richards. The business of free software: Enterprise incentives, investment, and motivation in the open source community. *Harvard Business School Working Paper Serie*, (07-028), 2006.
- [29] Chris Jensen and Walt Scacchi. Role migration and advancement processes in ossd projects: A comparative

- case study. In *ICSE '07: Proceedings of the 29th international conference on Software Engineering*, pages 364–374, Washington, DC, USA, 2007. IEEE Computer Society.
- [30] N. Jullien and J.-B. Zimmermann. New approaches to intellectual property: from open software to knowledge based industrial activities. In S. Labory and P. Bianchi, editors, *Industrial Handbook on Industrial Policy*, pages 243–264. Edward Elgar (EE), 2006.
- [31] Nicolas Jullien. Developing 'Free/Libre/Open Source Software (FLOSS)' - A Market Driven Investment. *SSRN eLibrary*, 2008.
- [32] Nicolas Jullien and Jean-Benoît Zimmermann. Firms' contribution to open-source software and the dominant user's skill. *European Management Review*, 6:130 – 139, 2009.
- [33] M. L. Katz and C. Shapiro. Network externalities, competition, and compatibility. *American Economic Review*, 75:3:424–440, 1985.
- [34] M. L. Katz and C. Shapiro. Technology adoption in the presence of network externalities. *Journal of Political Economy*, 75/4, 1986.
- [35] M. L. Katz and C. Shapiro. Systems competition and network effects. *Journal of Economic Perspectives*, 8(4):93–115, printemps 1994.
- [36] B. M. Kogut and A. Metiu. Open source software development and distributed innovation. *Oxford Review of Economic Policy*, 17(2), 2001.
- [37] H. Koski. OSS production and licensing strategies of software firms. *Review of Economic Research on Copyright Issues*, 2(2):111–125, 2007.
- [38] K. Lakhani and R. Wolf. *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects*. 2005.
- [39] R. N. Langlois and P. L. Robertson. *Firms, Markets and Economic Change, A Dynamic Theory for Business Institutions*. Routledge, Londres, New York, 1995.
- [40] Richard N. Langlois and Paul L. Robertson. Networks and innovation in a modular system: Lessons from the microcomputer and stereo component industries. *Research Policy*, 21(4):297–313, August 1992.
- [41] J. Ousterhout. Free software needs profit. *Communications of the ACM*, 42(4):44–45, April 1999.
- [42] Walt Scacchi. Free/open source software development: recent research results and emerging opportunities. In *ESEC-FSE companion '07: The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering*, pages 459–468, New York, NY, USA, 2007. ACM.
- [43] Sonali K. Shah. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, 52(2):1000–1014, July 2006.
- [44] W. G. Shepherd. *The Economics of Industrial Organization*. Prentice Hall International Editions, London, 1990. 3rd edition.
- [45] S. Slatter. *Gambling on Growth: How to Manage the Small High-Tech Firm*. Wiley, New York, 1992.
- [46] W. E. Steinmueller. The U.S. Software Industry: An Analysis and Interpretive History. In D. Mowery, editor, *The International Computer Software Industry: [A. 1996]*.
- [47] Rahdall Stross. *Planet Google*. Free Press, 2008.
- [48] D.J. Teece. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy*, 15 (7):285–305, 1986.
- [49] D.J. Teece, G. Pisano, and A. Shuen. Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7):509–533, 1997.
- [50] Laurent Thévenot. The plurality of cognitive formats and engagements: moving between the familiar and the public. *European Journal of Social Theory*, 10(3):413–427, 2007.
- [51] J. Tirole. *The Theory of Industrial Organization*. MIT Press, 1989.
- [52] Michaël Vicente. The impact of companies on the dynamic of open source software development networks: Convergence or differentiation of production models and labor division? doctoriales du GDR TIC et société, 2008.
- [53] E. von Hippel. Lead users: a source of novel product concepts. *Management Science*, 32(7), July 1986.
- [54] E. von Hippel. *The Sources of Innovation*. Oxford University Press, New York, 1988.
- [55] E. von Hippel. Open source software as horizontal innovation networks- by and for users. *MIT Sloan School of Management W.P.*, (4366-02), 2002.
- [56] E. von Hippel and G. von Krogh. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, 14(2):209–223, March-April 2003.
- [57] J. West. How open is open enough? melding proprietary and open source platform strategies. *Research Policy*, 32 (7):1259–1285, 2003.
- [58] J. West. The role of standards in the creation and use of information systems. Seattle, Washington, Dec. 13 2004. Standard Making: A Critical Research Frontier for Information Systems workshop.
- [59] Yunwen Ye, Kumiyo Nakakoji, Yasuhiro Yamamoto, and Kouichi Kishida. The co-evolution of systems and communities in free and open source software development. In Stefan Koch, editor, *Free/Open Source Software Development*, pages 59–82. Idea Group Publishing, Hershey, PA, 2004.
- [60] J.-B. Zimmermann. Le concept de grappes technologiques. Un cadre formel. *Revue économique*, 46(5):1263–1295, September 1995.
- [61] J.-B. Zimmermann. Logiciel et propriété intellectuelle: du copyright au copyleft. *Terminal*, 80/81:95–116, 1999.