Firms' contribution to open source software and the dominant skilled user

Nicolas Jullien, Jean-Benoît Zimmermann

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°2008-34

Firms' contribution to open source software and the dominant skilled user.

Nicolas Jullien
Jean-Benoît Zimmermann

Octobre 2008

Firms' contribution to open source software

and the dominant skilled user.

Nicolas Jullien – M@RSOUIN

Jean-Benoît Zimmermann – CNRS / GREQAM et IDEP

This version: September 2008 [*]

Abstract: *Free/libre or open-source software (FLOSS) is nowadays produced not only by individual benevolent developers but, in a growing proportion, by firms that hire programmers for their own objectives of development in open source or for contributing to open-source projects in the context of dedicated communities. A recent literature has focused on the question of the business models explaining how and why firms may draw benefits from such involvement and their connected activities. They can be considered as the building blocks of a new* modus operandi *of an industry, built on an alternative approach to intellectual property management. Its prospects will depend on both the firms' willingness to rally and its ability to compete with the traditional "proprietary" approach. As a matter of fact, firms' involvement in FLOSS, while growing, remains very contrasting, depending on the nature of the products and the characteristics of the markets. The aim of this paper is to emphasize that, beside factors like the importance of software as a core competence of the firm, the role of users on the related markets - and more precisely their level of skills - may provide a major explanation of such diversity. We introduce the concept of the dominant skilled user and we set up a theoretical model to better understand how it may condition the nature and outcome of the competition between a FLOSS firm and a proprietary firm. We discuss these results in the light of empirical stylized facts drawn from the recent trends in the software industry.*

1. Introduction

"Free/libre or open-source software" (FLOSS) is software whose source-code, that is the explicit expression of the programming work, remains openly accessible. Until recently, FLOSS was considered only to be of interest to programmers motivated by the building and sharing of a base of programs developed for their own needs (Lakhani & von Hippel (2003), Demazière & al. (2006)).

Today, open source software is increasingly integrated into many commercial offers (Novell, buying Ximian and SuSE, Sun open-sourcing its operating system, IBM open-sourcing its development tool software Eclipse, even Microsoft, which recently decided to distribute some of its software products under open license[1]). So Iansiti & Richards (2006) 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".

[*]The authors thank warmly the two anonymous referees for their excellent reports and suggestions and the EMR editors for the very accurate advice they gave us for the improvement of the paper. We are also indebted for helpful comments to the participants of the Conferences and seminars where we had the opportunity to present the project of this paper, more particularly the DIME London Conference "Intellectual Property Rights for Business and Society", Birkbeck University, 14th and 15th September 2006, the international conference "The diffusion of FLOSS and the Organization of the Software Industry: From Social Networks to Economic and Legal Models", Nice Sophia-Antipolis, May 31st, June 1st 2007, the Complex Markets Workshop, Warwick Business School, 31 March - 2 April 2008. This research has benefited from the support of the Institut Telecom (France), the French ANR and the European STREP "Complex Markets".

[1] http://solutions.journaldunet.com/0404/040407_microsoft.shtml

Lakhani and Wolf (2005), analyzing the results of a survey of 684 software developers involved in 287 FLOSS projects, show that “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 FLOSS project”. This situation, in which commercial business relies on the existence and durability of non-market activities, challenges traditional industrial economic theory.

As in any cooperative agreement devoted to technology or knowledge development, agents pool assets together in a “pre-competitive” phase and share the products of their efforts before returning to competition (Crémer et al. (1990), Bhattacharya and Guriev (2006)). But such agreements remain closed to third parties. On the contrary, a FLOSS project is an open contribution game in which the list of players is not bounded ex-ante by a cooperative agreement and the output of which is a public good that cannot be appropriated by any of the players in an exclusive way. This corresponds more to the formation of a consortium for the production of a standard[2].

In line with the seminal work of Teece (1986), FLOSS can be considered as an extreme case of “open innovation” (Chesbrough 2003), defined as “a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology” (Chesbrough, 2006). Dahlander and Wallin (2006) show that firms strategically sponsor individuals who occupy a central position in a community, in order to better access distributed skills and to control the direction of development of the related projects. Those firms seem to consider communities as “a complementary asset that exists outside firm boundaries and outside their ownership or hierarchical control”. This is even the sole control means when innovations, as observed by Von Hippel (2005), are produced by a community of innovative users that protect their innovation from private appropriation through a GPL license[3].

But things often stand differently. There exists a large variety of licenses that have been designed to achieve the co-existence of open source principles with private interests (Muselli, 2002, West 2003). But even in the case of GPL protection, intellectual property is not always shared among the members of a community. So, the database software MySQL is entirely owned by the eponymous firm[4]. Yet it has published its products under GPL license, thus abandoning any monopoly rent on them but allowing a free access for users and potential competitors.

So this is a radically brand new way of producing software that has emerged and become widespread during this last decade. It challenges the foundations and competition regime of the software industry, which was evolving more towards strengthening intellectual property protection (Bessen and Hunt, 2007). Two radically opposed ways of working are confronting each other, based on two opposing views about intellectual property management. On the one hand, the “traditional” position, referred to here as the “proprietary” approach, defends the need to strengthen the intellectual property regime to respond to a growing ease with which software codes can be appropriated and used without any fee[5]. FLOSS defenders, on the other hand, argue that that position is inefficient for innovation dynamics, reinforcing the market power of the sole dominant firms and thus the oligopolistic nature of the industry, which is economically inefficient[6]. The result of this confrontation is then of huge importance for the future of the software industry and, more widely, for the whole information and communication industry. From an even broader perspective, this case of open-source

[2] What we mean is that a player offers a standard by developing software that the other players can adopt and help to develop. This “unilateral” adoption is usually called ‘bandwagon’ in the literature on standards (see, for instance, Farrell and Saloner, 1985).

[3]The main principle of the GPL (General Public License) is to make its adopters disclose the source-code of the programs concerned and of any further improvement if they circulate them, as well as the free circulation of the code under the sole condition of maintaining its "open" character.

[4] This means that any developer wanting to make a contribution 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 a more classical closed license. So a customer that wishes not to reveal further enhancements of the source code has to pay for a closed license.

[5] See the arguments developed by the Business Software Alliance (BSA), which regroups the main actors of the software sector, such as IBM, Microsoft, Apple, Sybase, etc. on http://www.bsa.org/, especially their latest white paper (August 2008, http://www.bsa.org/country/Research%20and%20Statistics/~/media/96FC7EAFF3E84436AF62C3B393F207B1.ashx).

[6] See the arguments of the Free Software Foundation (http://www.fsf.org), or Boldrin and Levine (2007).

software can be viewed as a precursor of what could also take place in a wider array of industries where innovation has a strong role to play in competition and requires to access to a larger and larger range of knowledge and resources.

The outcome of this situation clearly depends both on the capability of software firms involved in FLOSS to set up business models likely to compete efficiently with those defending the "proprietary" position and on the inclination of firms incorporating software in their products either to consider FLOSS as an effective resource or even to support or contribute to their development. Currently, attitudes towards FLOSS appear to vary strongly according to the sectors of activities of what we shall refer as the information technology industry (IT industry in short).

The aim of this paper is to show that, beside factors like the importance of software as a core competence of the firm, one major explanation of such diversity can be found in the characterization of the users' skills according to the markets involved. In order to better understand this aspect, we introduce the concept of the dominant skilled user and we build a small competition model in which two firms, a proprietary one and a FLOSS one, choose their level of investment - and thus of product quality - and compete for market share. We show that the level of skill of the dominant users deeply conditions the nature and issue of the competition, as we shall illustrate through empirical stylised facts.

The paper is organised as follows. In section 2 we describe the structure of the IT industry and the variety of firms' attitude toward FLOSS. In section 3, we introduce the concept of the dominant skilled user and we present a formal competition model in which two firms, one open source and one proprietary, compete for a market characterised by a given distribution of users' skills. Section 4 provides a descriptive/empirical discussion of the application to FLOSS. Then we conclude.

2. The structure of the IT industry and the variety of attitudes toward FLOSS

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 the industry's borders and structure. However, the foundations of the industry have remained unchanged, since those described by Gérard-Varet & Zimmermann (1985), Zimmermann (1995) and Steinmueller (1996): 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 then possible to distinguish three large types of "vertical specialization": i) component producers; ii) computers and IT devices suppliers; iii) software editors and service companies providing applications.

The basic conditions for each kind of activity (characteristics of the products, of the users, hence of the demand) that shape the main aspects of the market structure have been extensively described in the literature (Genthon, 1995, Dréan 1996, Steinmueller, 1996). They help to explain the sources of competitive advantage and the conditions for a newcomer to enter a market, or for an incumbent firm to improve its market share.

i) Component producers supply the basic components of electronic devices like chips (Intel, Toshiba), hard disks or cards (ATI: graphic cards). The quality and performance of the main components built into an electronic device are, to an ever-increasing extent, considered as a critical feature of the performance of the product.

ii) Using these components, firms build machines more or less dedicated to specific uses. At one extreme, computers can be used for a wide scope of applications, depending on 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 applications[7].

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

–Servers are designed to manage, deliver and protect information on the networks. They must be high-performance, stable, but also compatible with network standards. An archetype of server producers is SUN.

–Microcomputers (with a growing market share for laptop computers) are used by end users, mainly as personal computers. In order to clarify the analysis, we have split this segment into the two categories of "*high quality computers" - HQC* - targeted at organizations or intensive end-users[8] and "*low price computers*" -*LPC* -, targeted at basic uses (Internet, for instance) and low-skilled users. Market structures and actors are dissimilar, even if some of the suppliers sell on both markets by differentiating their product ranges, in order to draw benefits from increased scale and scope economies. This appears quite similar to other industries like the automobile industry, where top- and bottom-of-the-range markets do not work similarly but can be partly held by the same manufacturers. An archetype of a "quality computers" supplier could be Dell or Toshiba, while an archetype of a "low price computer" could be Acer or Packard Bell.

–Dedicated machines such as PDAs, multimedia players, mobile phones, etc. are more and more present in information systems networking. Most of these devices are connected to the computer(s), but more and more also have access to the Internet (via Wifi, local networks or ADSL technologies), and share applications with the computer (e-mail, personal data management, etc.) Even devices such as video game consoles are increasingly connected to an Internet-based network. Archetypes of this evolution are Nokia's mobile phone, Nintendo's game console, Apple's music player (Ipod) or Palm's PDA.

iii) 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). Three typical strategic orientations can be described that have successively emerged in two waves. These three strategic options gave rise to a market segmentation that corresponds to different degrees of user sensibility to price, variety and specificity of the offer.

- **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 installed and adapted to their specific needs and constraints, with the help of service enterprises accredited by the software editor (Horn 2004, 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 firms and, more recently, outside, with the diffusion 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:

- **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 this latter in 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,

[8] People playing games, watching video films on their computer or using it for high performance professional purposes.

after other takeovers, to enlarge its applications portfolio[9]. Another illustration is given by Symbian in the field of operating systems (OS) for mobile applications.

- **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 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 at 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). One single segment (the platform market, when looking at the OS for PC computers market) can be considered as a quasi-monopoly with very high barriers to entry, controlled by a dominant leader (Microsoft, Oracle) that intends to use this dominant position in order to extend its dominance over related application tools (See, for example, Gartner for a description of the database market and its level of concentration[10]).

In such oligopolies, differentiation strategies generally play an important role. In the sole segment of low-price computers, where competition is mainly based on prices, the strategies are based on the exploitation of new market potential through vertical differentiation with the traditional PC market. In the other segments, strategies are rather of horizontal differentiation, either related to the integration of new features and high performance tools or to market segmentation through hard-soft-content bundling (mainly based on proprietary standards).

So new entries require either better performance/cost ratios (for instance lower price computers or better computational capacities for servers or high-quality computers) or innovative horizontal differentiation (e.g. new features in personal communication tools or business software).

Last but not least, horizontal differentiation can also target a sub-group of users with specific needs not fulfilled by the existing offer, as Samsung did when proposing a fold display mobile phone. This is also a possible strategy in the business software market (to propose a targeted offer at the best possible price to a “niche” market).

Since the mid-1990s, a growing number of commercial firms, both new entrants and incumbents, have decided to integrate FLOSS products into their own specific offer or toolboxes, even investing by different means in FLOSS development. Of course, these new emerging strategies must be understood in the light of the IP protection practices prevailing in each market segment and the need to strengthen competitive advantages or to rely on new ones.

Regarding the degree of involvement in FLOSS dynamics, the most active actors seems to be found in sectors where software development and use is either a core activity or a crucial condition for performances, as is the case for server manufacturers or architects (e.g. the adoption of Linux by IBM and HP since the beginning of the 2000s). At the other extreme, the weakest involvement is found amongst hardware suppliers that can only feel concerned by FLOSS for compatibility and price purposes.

When FLOSS adoption is related to marginal aspects of differentiation, it seems to have little impact on industrial structure and competition. This is generally the case for most hardware producers, when hard-soft-content is no longer bundled (servers, computers, PCTs, DVD and MP3 players, etc.)

[9] “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

[10] http://www.gartner.com/it/page.jsp?id=507466

Regarding software firms, their core competences have evolved and shifted significantly. Their main challenge is less and less to supply a "software solution" to a given problem at a given time, but increasingly to deal with short to long term uncertainty over IT system production and management. Users ask for solutions to protect them against uncertainty, granting interoperability, bug resolution, the satisfaction of new needs and the integration of technical advances. The trade-off between available solutions is not posed in terms of their cost of acquisition but of their "TCO" (total cost of ownership), in which the future costs and the costs for granting interoperability and adaptability have to be estimated. This is precisely what architects, business programs and platform producers sell to skilled users, aware of these problems and signals. On these markets the FLOSS organization seems to represent an asset for producers, who can display their involvement and succeed in building sustainable business models (see the examples of RedHat, MySQL or, in France, Linagora). But, as explained before, this is only an asset if the market regards FLOSS as providing a value-added to the product, i.e. if it brings the users a potential for increasing their utility.

To a large extent, the differing degrees of involvement in FLOSS between different firms can be explained by the position of software in the firm's business model (as core activity or not, see Dahlander and Magnusson, 2005) but also by the role played by skilled users in the selection of the offers on the market. If the first aspect is clearly connected to a well-developed recent literature, the second remains rather new and relatively unexplored. From that stems the need to better understand the role of the users and the way it may impact the firm's ability to achieve a viable business activity capable of competing efficiently with proprietary software firms.

To this end, we introduce in the next section the notion of the dominant user's skill and we present a theoretical competition model in which two firms with contrasting attitudes toward intellectual property protection compete on a market characterized by a given distribution of users' skills. We will see how this distribution may drive both the nature (price vs quality) and the outcome (market shares and profit levels) of the competition. For this purpose we shall give a formal conceptual base to the notion of dominant user's skill.

3. A competition model with customer feedback

The users, understood as the persons choosing the solution (thus not always being the "end-users"), are rather different from one market to another, causing the competitive advantage to rely on different features.

Let us distinguish between three main types of users according to their relation to the product and the technology (Zimmermann 1995, Kogut and Metiu 2001, Von Hippel 1988, 2002). The first is the category of "Naïve 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. The second is the category of "Kogut-Metiu Users" (KM)[11] who are not able to contribute to software development but can generate new features or innovations by revealing their own needs. Above all, they represent an irreplaceable testing and debugging base. KM users are sensitive to price and quality arguments. The third category is that of the "Von Hippel Users" (VH) who act as "sources of innovation" (Von Hippel, 1988) 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. So, even if they are VH or KM types, individual users in such a market can hardly make their views heard (in terms of technical choice, for instance). 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

[11] In reference to the notion of "frontier-users" put forward by Kogut and Metiu (2001)

distribution by his own means (local government for secondary schools in France[12], education in rural areas in developing countries[13], etc.) 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. Our aim here is to characterize this distribution under the assumption that it usually takes a unimodal shape, meaning that most of the users in the market segment are concentrated around a given level of skill, corresponding to what we call the dominant user's skill.

In this model presented below, we consider a competition game between a proprietary and an open-source software supplier on a market characterized by a given distribution of users' skills.

The users

In a pretty standard way, we consider each individual user to be characterized by a given level of preference for quality $\theta \in [0,1]$[14]. In terms of the problem addressed here, this level can be considered as a good proxy of the individual user's skill: information technology products are endowed with a wide usage potential that is actually exploited more or less fully, according to the user's ability. This ability is tightly correlated with the user's level of skill in computing: the more skilled, the more he is able to benefit from the technical quality of the product.[15]

So for a user of level θ , the utility obtained from a product of quality *s* sold at price *p* can be written as

$$U = \theta s - p \text{ if he buys the product;}$$

$$= 0 \quad \text{otherwise.}$$

It is considered that the weight of the user population is normalized to 1.

A given market is then characterized by the distribution of preferences given by a function

$$f: [0,1] \to R_+ : \quad \int_0^1 f(\theta) d\theta = 1 \qquad (1)$$

The suppliers

Let *l* denote the open-source firm and *m* the proprietary firm.

Both sell a substitutable product vertically differentiated through a quality level s > 0.

The proprietary firm chooses the quality level of its product between a low and a high level

$$s_m \in \{s_L , s_H\} \text{ with } 0 < s_L < s_H$$

It has to pay a fixed cost

$$c_m \in \{c_L , c_H\} \text{ with } 0 < c_L < c_H$$

The open-source firm and the users' feedback effects

[12] With the aim to provide "a computer for each pupil": http://www.ordina13.com/, http://www.ordi35.fr/

[13] See, 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

[14]See Tirole J. (1988), Chapter 2.

[15]See for instance Blili, Raymond and Rivard (1998)

The quality s_l of the product delivered by the open-source firm depends on both the level of contribution c_l of l to its development and the amount of the feedback effects from its user base, so that

$$s_l = q\,(c_l\ ,\ \tau_l) \qquad (2)$$

where τ_l is the users' feedback effect, measured as the aggregated skill level of the user base for the open-source product;

$$\tau_l = \int_{\Theta_l} \theta\, f(\theta)\, d\,\theta \qquad (3)$$

where Θ_l is the user base of l , assumed to be compact.

q is assumed to be twice differentiable increasing and quasi-concave in c_l and τ_l and $q(0,.) = s_d$ where s_d is the standard quality of the product as freely available on Internet.

A simple expression of q is given as linear in c_l and τ_l by

$$s_l = \lambda\ c_l\ (1+ \tau_l\ ) \ (4)$$

so that $s_d = 0$ in so far as when it invests 0, l can only sell what it takes from the community and doesn't provide any added value in terms of interface, adaptation or services that could justify a positive price $p_l > 0$.

In order to avoid a trivial situation, λ is assumed to have a value such that the open-source firm needs to invest more than the proprietary firm to reach the same levels of quality $s_L < s_H$, so that

$$\lambda < \mathrm{Min}\ (s_L\ /\ c_L\ ,\ s_H\ /\ c_H\ ) \qquad (5)$$

Proposition 1: **At the equilibrium,**

if $s_l < s_m$ then for any user of level θ choosing the open-source software, ∀ θ' < θ , θ' then chooses open-source software

if $s_l > s_m$ then for any user of level θ choosing the open-source software, ∀ θ' > θ , θ' then chooses open-source software

The proof of proposition 1 is given in the appendix.

Corollary:

Let $\tilde{\theta}$ denote the value of θ at which a user has no preference between l and m

$$\tilde{\theta} = \frac{p_m - p_l}{s_m - s_l} \qquad (6)$$

and θ_0 the minimal level of the users entering the market.

When $s_l < s_m$, $\theta_. = \frac{p_l}{s_l}$ then $\Theta_l = [\theta_0\ ,\ \tilde{\theta}\ ]$ and $\Theta_m = [\ \tilde{\theta}\ ,\ 1]$

When $s_m < s_l$, $\theta_. = \frac{p_m}{s_m}$ then $\Theta_l = [\ \tilde{\theta}\ ,\ 1]$ and $\Theta_{lm} = [\theta_0\ ,\ \tilde{\theta}\ ]$

The game:

m chooses to invest $c_m = c_L$ resp. c_H for a quality $s_m = s_L$ resp. s_H and sets its market price p_m

l invests c_l and gets a quality level s_l at price p_l , with s_l either lower or greater than s_m .

Respective profits can be written $\Pi_m = p_m\ |\Theta_m| - c_m$ and $\Pi_l = p_l\ |\Theta_l| - c_l$ as defined above.

So the game is the following:

$m \setminus l$	$<$ sm	$>$ sm
s_L	$\Pi_m^{\underline{L}}$, $\Pi_l^{\underline{L}}$	$\Pi_m^{\bar{L}}$, $\Pi_l^{\bar{L}}$
s_H	$\Pi_m^{\underline{H}}$, $\Pi_l^{\underline{H}}$	$\Pi_m^{\square}$, $\Pi_l^{\square}$

The reasoning is backward:

If m chooses L, l then chooses the low strategy ($s_l < s_m$) or high strategy ($s_l > s_m$) that brings it the higher profit ($\Pi_l^{\underline{L}}$ or $\Pi_l^{\bar{L}}$) at the equilibrium. The m profit Π_m^L follows. In the same way, if m chooses H, there follows a profit Π_m^H for m. So m will choose the strategy L or H that brings it the higher profit, and so does l.

Of course the complete resolution of the game is very heavy in the general case, due to the nature of the indifference level $\tilde{\theta}$ that is defined through an implicit function. That's the reason we prefer to develop an approach aiming to characterize the different situations in the different quadrants of the game, and the conditions relating to the transition from one quadrant to another.

<u>Definition</u>: The dominant user's skill

As seen before, each market in the software industry is characterized by a dominant type of user that we have described through a typology as “naive”, “K.M.” or “V.H.” according to their skill level in relation to the software product. Of course, this dominant type coexists with users of different skill levels, but the distribution can be considered as relatively unimodal. Hence, we characterize the level of the dominant user not through the mean value of θ among the population but through a stronger **criterion of stochastic dominance,** meaning that if f and g are the distribution of users' skills characterizing two markets, then

$$g \gg f \quad \Leftrightarrow \quad \forall \ \hat{\theta} \in [0,1[\ , \ \int_{\hat{\theta}}^{1} \theta\, g(\theta)\, d\theta > \int_{\hat{\theta}}^{1} \theta\, f(\theta)\, d\theta \qquad (7)$$

This criterion expresses a shifting of the mass of the users from a lower to a higher level of skills.

In order to understand how and why the dominant user's skill influences the level of investment of the open-source firm, we study two polar cases. The first (North-West quadrant) is characterized by a low skill level of the dominant user, when a large majority of users are concentrated at a low level of skills not far from 0. The opposite polar case (South-East quadrant) occurs when the population of users is dominated by highly-skilled users.

North-West Quadrant: $s_m = s_L$ and $s_l < s_m$

In such a situation, m chooses a low quality while l cannot rely on a potential of users' skills large enough to reach a quality level s_L without investing in a non-profitable way (assumption (5) on λ value). Then l can only aim at a low level of quality and a low price, targeting a demand that is more sensitive to price than to quality.

Proposition 2:

There exists a low threshold of the dominant user's skill under which the open-source firm invests little, and stays at a low level of quality, targeting a relatively price-sensitive market.

Proof and technical remarks about proposition 2 are given in the appendix.

The transition between the two polar cases (NW and SE quadrants) when the dominant user's skill is increasing can take two different paths: NW→ NE → SE or NW→ SW → SE, depending on the values of the parameters.

South-East Quadrant: $s_m = s_H$ and $s_l > s_m$

This quadrant corresponds to a high skill level of the dominant user, when a large majority of users are concentrated at a high level of skills not far from 1. In such a situation, m will choose a high quality, while l can invest in a profitable way by relying on a potential of users' skills large enough to reach a quality level $s_l > s_L$.

The limit condition is obtained for a symmetrical duopoly situation equivalent to that of the NW case (8) with θ_0 root of $2\, c_H\, \theta_0 = s_H\, (1 - F(\theta_0))$

and

$$\frac{1}{2}\int_{\theta_0}^{1} \theta f(\theta)\, d\theta = \frac{s_H}{\lambda c_H} - 1 \qquad (11)$$

From this stems the following proposition.

Proposition 3:

There exists a high threshold of the dominant user's skill above which the open-source firm invests a large and increasing amount, obtains a high and growing level of quality and makes a profit that increases with the level of the dominant user's skill.

The proof of Proposition 3 is given in the appendix.

In conclusion, we can say that if the dominant user is unskilled, and so price-taker, FLOSS may be a good strategy if it enables the firm to supply a product at a lower price than the proprietary one, but requiring a low level of investment to adapt or improve the FLOSS solution. In the opposite situation, when the dominant user is very skilled, an offer based on FLOSS and a high level of complementary services can succeed. This result is in accordance with the work of Henkel (2006), who shows in a duopoly model that "a regime with compulsory revealing can lead not only to higher profits, but also to higher product qualities than a proprietary regime". In the intermediate situations, the FLOSS strategy may be less efficient than the proprietary one.

4. Discussion: The critical importance of the users' skills

How and why may those different users contribute directly or indirectly to FLOSS projects? First of all, contribution does not necessary imply code development but can take various forms in the product development and improvement. Users have to be considered as valuable "sources of innovation" (Von Hippel), not only for testing and debugging programs but also for improving product usability and performances. People decide to contribute if they get interested by the product, or if they have a problem, in which case they can either report the problem directly (VH users) or through an intermediary, the supplier for instance, that allows the user to pass from a passive to an active use of the project.

Empirical observation of firms' involvement in FLOSS development is consistent with the results of the model. In fields where the dominant users' skill is either high or very low, firms invest in FLOSS. When the dominant user's skill is intermediate, the dominant design remains that of the classical proprietary model.

Table 1 below summarizes the main types of users likely to be found in each sub-sector of the IT industry.

Table 1. The dominant user type in each IT sub-sector.

Actors/ products	***Dominant user type***	***Comments***
Components	VH	Component producers supply hardware manufacturers, aware of the quality and quality-price aspects of the components they will use, as well as the effects of brand reputation of these latter as a signal of quality for their own products.
Servers	VH	The clients are computer-literate people, able to express needs in technical terms, to develop software for their own needs, and to innovate by themselves.
High Quality Computers	KM	HQC users are somewhat less computer-literate than server users; they can be characterized as "intensive frontier users". So the market is looking at a good performance-to-price ratio.
Low Price Computers	N +. KM	LPC is a mass market; users have no particular skills except in the case of intermediation by a "prescriber".
PCT	N + KM	PCT and players are relatively mass markets, but some advanced users (more in the PCT field and particularly in the PDA market) can play a constructive role in the development of new features.
Players	N	
Platform producers	KM + N	For the OS, as for hardware components, most of the end-users buy a computer with an OS already installed. So the actual users in our sense of the term are computer manufacturers, service companies and sophisticated end-users capable of installing an alternative operating system for their proper use or the use of their customers. On other platforms (database, middleware), the users are also computer manufacturers, service companies and highly-skilled users.
Business solution producers	VH/KM depending on the markets	In the business solutions market, users are professionals. They are able to make a technical evaluation of the product, to carry out trials and tests. This means that people may have skills in the functional domain (what they want, how the software works), and sometimes in the technical one (able to adapt or develop software to meet their own needs, especially in the tools for computer professionals market).
Architects	N (+VH)	Large firms and organizations include very sophisticated users (IT division). SMEs or corporate divisions, at local or sectorial level, are clients of very heterogeneous but rather low IT skills. However, clients may be quite precise in the definition of the services they need, and so in the specification of the application characteristics.

1. In the **servers market**, producers have habitually provided proprietary solutions with proprietary Unix[16]. Here, suppliers are dealing with highly-skilled VH clients that can make an essential contribution in the context of FLOSS opening. The rise of PC servers has permitted some users to avoid such a bundling problem; moreover, using Linux allows a cheaper offer (vertical advantage) reusing Unix programs (content) portfolio. Thus, some firms have been able to widen the servers market from VH users capable of managing their systems by themselves to KM clients, sensitive to prices, but also to the quality of a PC server fitted out with Linux. So new entries have been experienced like the Cobalt[17] one, but the main actors of the Unix "world" have also rapidly developed their own offers, cutting down the sources of vertical differentiation[18].

2. In the segment of **low price computers** (LPC) where naïve clients are the driving force behind demand, competition is overall based on prices. Installing a FLOSS OS can be considered a way to reduce prices and hence to extend the firm's market share to a wider market, following Asus' "eee PC" offer, or at least to enable a new entry by compensating for not-yet-accessible economies of scale[19].

[16] See West (2003) for a full discussion of FLOSS strategies in that sector.

[17] Cobalt was bought by SUN, which dissolved the products into its own offer. See http://www.sun.com/hardware/serverappliances/eol.html

[18] It is worth noting that SUN, on the contrary, being the leader on the UNIX market, has been reluctant to adopt Linux and is today the server constructor which has the most difficulties in adapting its business model, with recurrent losses.

[19] Since the middle of 2007, Dell proposes Ubuntu Linux distribution on one of its lowest-price laptops (http://www.dell.com/content/topics/segtopic.aspx/linux_3x?c=us&cs=19&l=en&s=dhs), and since 2007 Asus has been selling an ultra-portable computer running Linux, called "eee PC", at less than €300. This has been the "most wanted 2007 Christmas gift", according to the constructor, http://eeepc.asus.com/global/.

3. Between these two cases there is the **high quality computers** (HQC) market. HQC producers may find it hard to switch from Windows to Linux, because this would mean either acquiring new skills (OS management and improvement), or sub-contracting this maintenance to Linux editors (e.g. RedHat or SuSE), which may lead to another dependence and to difficult relations with the dominant provider. Nevertheless, a possible future evolution in this sense is likely to arise from the pressure of customers becoming more aware of the potentialities of switching to FLOSS, as illustrated by Dell's move towards the provision of Linux-based computers. In the near future, most of the HQ will probably switch to debundling their machines from the associated OS, to give KM users the choice between a Windows and a Linux platform, or to segment more their offer between VH users with the Linux offer and KM users with Windows. It is worth noting that in this desktop market, the main push in favour of open source, for the time being, is driven by organizations or institutions (which we consider as VH users) that take decisions to equip a large number of end-users. Examples are the French “Assemblée Nationale” (French Congress) that has contracted with a service company to install Linux on all the computers provided to MPs[20], or the initiatives of the Nigerian[21] and Macedonian[22] governments for schools.

4. **Personal communication tools** represent another intermediate case with less skilled customers (KM+N) and a weak degree of involvement on the part of commercial actors only motivated by preoccupations of compatibility and absorptive capacity. There are lots of FLOSS products for PCT, or Mobile Computers[23]. Some are proposed by VH users, other by the constructors. At one extreme, in the games consoles segment but also to a lesser extent in the music player market, proprietary formats have introduced, as seen above, a strong bundle of hardware-software-content. Thanks to the MP3 standard or new existing or emerging open standards like Ogg, new entries are always possible in segments like the music players market, but the main actors, like Apple, remain on a strict proprietary strategy. On the contrary, barriers remain high on the video game players market due to the scarcity of independent games capable of running on Linux, unlike the PS2, Xbox and other proprietary standards games. Moreover, when they exist, such games seem harder to obtain for simple users. On the contrary, PCT suppliers have begun experiments with FLOSS products over the last couple of years:

- if the leader, Nokia only sold an Internet tablet based on Linux and a development community[24], there are lots of open-source projects around Symbian (partly owned by Nokia, partly by Sony-Ericsson)[25], mainly dedicated to tools for developing applications (libraries, development tools, etc.) and Samsung proposes the first smart phones based on Linux[26];

- the PDA Operating system editor Palmsource is working on the integration of its product on a Linux kernel[27].

For the same reasons as for PC computers, we hardly see naïve or KM people switch from an installed operating system to a FLOSS one. So constructors will continue to drive the market and decide what they integrate in their offer. But operating systems are not at the heart of the product differentiation, which is more based on ergonomic aspects and hardware characteristics. In the absence of an established *de facto* standard, as it stands in the PC market, Linux is to be considered by PCT suppliers, as it is free of charge and benefits from a community of developer-users capable of developing new features and new products outside of any proprietary control. Implementing Linux on PCT devices may appear as a good strategy to limit differentiation to the core competences of the manufacturers. For firms like Nokia, who do not own or control any operating

[20] http://www.zdnetasia.com/news/software/0,39044164,61970345,00.htm

[21] http://www.zdnet.co.uk/talkback/0,1000001161,39290511-39001070c-20088736o,00.htm .

[22] http://www.linuxtoday.com/infrastructure/2007091902626NWDPPB

[23] See, for instance, http://tuxmobil.org/ a web site dedicated to Linux and mobile computers.

[24] Nokia 770 Internet Tablet: http://www.nokiausa.com/770/1,7841,feat:1,00.html. Development community: http://www.maemo.org/

[25] In June 2008, Nokia announced to be acquiring the whole share of Symbian and open source it under Eclipse license. See the Symbian foundation Web site: http://www.symbianfoundation.org/

[26] http://linuxdevices.com/news/NS2854558742.html

[27] Palm and Linux: http://news.com.com/2102-1041_3-6175171.html?tag=st.util.print. The web site dedicated by Palm to open source: http://www.palmsource.com/opensource/.

system, there are strong incentives to use FLOSS. Palm is also a good example of a company that is now turning toward Linux, after having sold its OS division.

5. In the **software platform market**, the Linux distribution market is another very good illustration of the key role of the demand. Linux publishers, like RedHat, SuSE, Mandriva (formally Mandrakesoft), have been among the first commercial actors to enter the market using FLOSS. This could appear to be obvious on a mass market with rather naive users and a significant price-based competition. But today, the retail store sales of OS packages represent a negligible part of the revenue of such firms[28], and a major part is targeted at the business market.

One might explain this by the development of broadband connection, thanks to ADSL. But we believe a more important explanation lies in the skills of the users and the construction of the offer. Consumers buy computers with an OS already installed and few of them are skilled enough to install a different one. Additionally, there are no incentives to do so because the pre-installed OS has already been paid for with the computer. So the diffusion of FLOSS OS on desktop/laptop PCs depends more on the strategies of constructors, as discussed above, than on direct installation by users. In other words, what needs to be taken into account is the possible changing attitude of computer producers towards the dominant position of Microsoft in the OS market. For VH people wanting to install Linux on their PC, other, more technically oriented distributions exist, like Debian, and there is no need to pay for these distributions, available for download on the Web.

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 questions involved in installing and configuring an OS. It is also easier to buy a machine without an operating system installed, and the relative price of the OS is lower. FLOSS gives these users access to a cheaper, more open and more adaptable Unix-like operating system than they could find in the traditional Unix offer. This gave FLOSS OS publishers an undeniable competitive advantage, at least until server constructors started to offer PC servers with Linux.

6. In the **business software market**, the more skilled the users are in terms of software development skills and (although this is a lesser driving force) in terms of expressing their functionality requirements, the more FLOSS concepts and related industrial offers are likely to spread.

It is clear that the use of open-source business software, enabling savings on 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. It may even be compulsory when dominant players already exist on the market (such as the database market where MySQL proposes software products competing against those of Oracle, IBM and Microsoft, who represent more than 80 % of the market) or when customers are highly sensitive to price (such as the ERP market which increasingly concerns SMEs and where open-source products like ERP5 or tiny ERP are now available). This strategy also enables the association of a corporate brand with a product, therefore increasing the notoriety of the firm through distribution of the latter. Moreover, on these technical markets, especially when the customers are developers, availability of the code promotes cooperation. 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, MySQL and some small service 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 Von Hippel 1988).

Obviously, capitalizing on existing products is more difficult, even if, as Muselli (2002) explained, with entire control of the software, a dual license strategy can be set up to sell the program when requested by customers (because, for example, they want to integrate it in a larger, closed package). This is what companies like Qt or MySQL offer. But, today, the main source of revenue again comes from services, more precisely what we call

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

the “3A services” (assistance, assurance and adaptation to use). Otherwise, adaptation services must be significant enough to finance development of the product. Therefore, the objective is to transform a handicap (significant investments) into a commercial advantage, by increasing the business feedback from users and by considering openness as a way to reduce transaction costs and as a signal of quality. 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).

This explains why open-source business products are developed mainly in “business” software (ERP, computer infrastructure software such as compilers), where users ready to pay for configuration, maintenance or assistance services are numerous. But the scope could easily extend to many technical/professional software activities.

7. As far as the **architects market** is concerned, 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 therefore be available in the form of open-source software and legally modifiable.

The competitive advantage in using free software, 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 he keeps 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 asset, and incumbents, such as IBM for its service activities[29]. Traditional service firms like Cap Gemini are more agnostic with regard to the technologies 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 organisations, skilled computer users that are receptive to the opportunity to integrate the most advanced software components, developed under open licenses. So they are becoming increasingly involved in FLOSS as the market grows and matures [30].

5. Conclusion

In application markets, all the recent new entries have been based on the competitive advantage drawn from the FLOSS label: FLOSS OS publishers (like RedHat), FLOSS database producers (MySQL), FLOSS service companies (VA Linux, or Linagora in France).

Today, incumbents are also adopting this strategy (IBM with Eclipse, SAP opening its database system, even Microsoft opening some of its technical tools.) This could shortly become the benchmark of industrial organization on these markets, inducing a growing control of FLOSS development by commercial firms and a spectacular extension of open IP regimes in the software field.

In such conditions, the open IP regime can be seen as a very efficient solution to the Schumpeterian dilemma, insofar 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.

---

[29] As explained by Slatter (1992), one of the main strategies for newcomers in technological markets is technological differentiation. Basing its offer on new FLOSS products can be seen as a way for new service companies to differentiate.

[30] In 2005, Gartner forecasted that « 2008, 95 percent of Global 2000 organizations will have formal open-source acquisition and management strategies » (http://www.gartner.com/DisplayDocument?doc_cd=125868)

Nevertheless, 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. This is the core of the open-source model and, as seen in the paper, the recent move towards commercial structuring doesn't contradict this foundation. But, as seen before, it is clear that things function differently according to the place and type of the users in the different markets of the IT industry. As we have stressed, users play a critical role in the design of FLOSS strategies in the various market segments, both in the formation and specification of the demand and in their capacity to produce efficient feedbacks that contribute to raising the quality level of the products.

For a better understanding of how this works in a competitive context, we have developed a small competition model in which one open-source software provider competes with one proprietary software firm. We have focused on two polar market situations driven by contrasting distributions of skills among users. At one extreme, users are concentrated at a very low level of skills. There are no significant feedback effects and the open-source software firm, like the proprietary one, remains at a low level of product quality, is little involved in product development and finds a competitive advantage in terms of price. At the other extreme, the more skilled the users, the higher the quality level of the open-source product, the more the open source firms invests and makes profits. The two opposite situations show how involvement of commercial firms in FLOSS is not contradictory with the essence of open-source rationale and can combine with the way communities operate to achieve the viability of an alternative economic model based on open knowledge.

The FLOSS movement has sometimes been presented as a canonical model of production for the open innovation paradigm, 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. 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 of the evaluation, by taking into account the impact of the users in these productions, and their bargaining power.

.Appendix

- **Proof of the proposition 1**

θ adopts the open source software if and only if $U_l(\theta) > U_m(\theta)$

$$\Leftrightarrow s_l\,\theta - p_l > s_m\,\theta - p_m$$

when $s_l < s_m$ (and $p_l < p_m$)

$$\Leftrightarrow \theta \leqslant \frac{p_m - p_l}{s_m - s_l}$$

which remains true for any $\theta' < \theta$

and symmetrically for $s_l > s_m$

- **Proof and technical remarks about the proposition 2**

The situation described in the N-W quadrant will hold as long as l does not catch up m at level s_L, then as long as $\lambda\, c_l\,(1+\tau_l) < s_L$.

At the limit, when $s_l = s_L = s_m$, the two firms have to sell at the same price and profit 0, sharing the market equally.

So the demand for each of the firms can be written

$$d = \frac{1}{2}\int_{\theta_0}^{1} f(\theta)\,d\theta = \frac{1}{2}[1 - F(\theta_0)] \qquad (8)$$

where $\theta_0 = s_L / p$ and $p = c_L / d$ (the profit of m is equal to zero)

so as θ_0 is the root of $2 c_L \theta_0 = s_L (1 - F(\theta_0))$.

The profit of l is equal to zero so that $\Pi_l = d\,p - c_l \Leftrightarrow c_l = c_L$

The condition $s_l = s_L$ can then be written as a condition on the users feedback

$$\tau_l = \frac{1}{2}\int_{\theta_0}^{1} \theta f(\theta)\,d\theta = \frac{s_L}{\lambda c_L} - 1 \qquad (9)$$

Where θ_0 has been defined above.

Of course, we have only focused here on the limit conditions.

Note that (s_L , s_L) is never a Nash stable equilibrium. At the equilibrium l always chooses $s_l < s_L$ so as to avoid Bertrand competition.

For a given distribution f of the users' skills, if l has only quality aims, it can reach

$\bar{s}_l = Max_{\Pi_l \geqslant 0}\, s_l$ so that $\Pi_l(\bar{s}_l) = 0$.

When l, as commercial competing firm, aims to maximise its profit, it chooses

$$s_l^* = \text{Arg Max } \Pi_l\ (s_l) / s_l \in]\,0\,,\ \bar{s}_l\,[$$

With a critical distribution f_c when l choosing $s_l = s_L$, neither l nor m would make any profit.

So to maximize its profit level, l will choose $s_l^* < \bar{s}_l^c < \bar{s}_L$, so that l and m make positive profits.

Depending on the value of λ, the above threshold can be more or less high. It is decreasing with λ and c_L and increasing with s_L . It is then also possible that m decides to choose the high quality before l has been capable to catch up. This will happen as soon as the expected profit for m with a low strategy for l is greater for a high than for a low quality.

The limit condition is obtained with equal profits, so that

$$[1 - F(\tilde{\theta}_L)]\,p_L - c_L = [1 - F(\tilde{\theta}_H)]\,p_H - c_H \qquad (10)$$

which also depends on the differential $c_H - c_L$.

- **Proof of the proposition 3**

When f dominates the condition (11) above, l and m target high quality products. So m chooses $\mathbf{s_m} = s_H$ at cost c_H and l invests c_l and targets a quality $s_l > s_H$ with $s_l = \lambda\, c_l\, (1 + \tau_l)$

where $\tau_l = \int_{\tilde{\theta}}^{1} \theta f(\theta)\,d\theta$ and $\tilde{\theta} = \frac{p_l - p_m}{s_l - s_H}$

At prices p_l and p_m are such that $p_l > p_m$. The payoffs can then be written

$$\Pi_l = [1 - F(\tilde{\theta})]\,p_l - c_l \text{ and}$$

$$\Pi_m = [F(\tilde{\theta}) - F(\theta.^m)]\, p_m - c_H \quad \text{where} \quad \theta.^m = \frac{p_m}{s_H}$$

We assume that the user base improves its aggregate skill level following the criterion of stochastic dominance (7) shifting from a distribution f to a distribution g such that

$$\forall\ \hat{\theta} \in [0,1[\ ,\quad \int_{\hat{\theta}}^{1} \theta\, g(\theta)\, d\theta > \int_{\hat{\theta}}^{1} \theta\, f(\theta)\, d\theta$$

The consequences of such a shift is above all and by definition a growth of the users' feedback from any population $[\hat{\theta}, 1]$, $\forall\ \hat{\theta} \in [0,1[$. Improving s_l tends to lower, ceteris paribus, the level of the indifferent user $\tilde{\theta}$, enlarge the l market base, increase its profit and enable it to lower its price p_l and to reinforce its level of investment c_l then reinforcing again the quality level s_l until reaching a new competitive equilibrium $(s_l{}^g, s_H)$ where $s_l{}^g > s_l$.

.References.

J. Bessen and R. M. Hunt, 2007, “An Empirical Look at Software Patents”, Journal of Economics and Management Strategy, Volume 16, Number 1, Spring, 157-189.

S. Bhattacharya and Guriev S, 2006. 'Patents vs. trade secrets: Knowledge licensing and spillover.' *Journal of the Economic Association* 4, no. 6, pp. 1112-1147.

S.Blili, L.Raymond, S.Rivard, 1998, “Impact of task uncertainty, end-user involvement, and competence on the success of end user computing”, Information Management **33** (3) 137-153.

Boldrine M. and Levine D.K., 2007, “Open Source Software, Who Need Intellectual Property?”, *The Freeman, Ideas on Liberty*, Vol. 57, N°1, January, Foundation for Economic Education.

H. Chesbrough, 2003. Open innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press, Boston MA.

H. Chesbrough, 2006. Open innovation: a new paradigm for understanding industrial innovation. In: Chesbrough, H., Vanhaverbeke, W., West, J. (Eds.), Open Innovation: Researching a New Paradigm. Oxford University Press, Oxford.

J. Crémer, d'Aspremont C. and Gérard-Varet L.-A., 1990. “Incentives and the Existence of Pareto-Optimal Revelation Mechanisms”, Journal of Economic Theory, vol. 51, n. 2, August, pp. 233-254.

L. Dahlander and M. W. Wallin, 2006**.** A man on the inside: Unlocking communities as complementary assets. Research Policy, 35, 1243-1259.

L. Dahlander and M. G. Magnusson, 2005**.** Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms. Research Policy, 34, 481-493

J. De Bandt, 1995. Services aux entreprises: informations, produits, richesses. Economica, Paris.

D. Demazière, F. Horn and N. Jullien, 2006, How free software developers work. The mobilization of “distant communities”, working paper, M@rsouin. http://www.marsouin.org/articles.php3?id_article=116. A former version (in French) has been published in Cahier lillois d'économie et de sociologie, 2005, n°46, pp 171-194.

G. Dréan, *1996. L’industrie informatique, structure, économie, perspectives. Masson, Paris.*

J. Farrell and G. Saloner, 1985. Standardization, compatibility and innovation. *Rand Journal of Economics* 16 , pp. 70–83.

C. Genthon, 1995. Croissance et crise de l'industrie informatique mondiale. Syros, Paris.

L-A. Gérard-Varet and J-B. Zimmermann, 1985, Concept de produit informatique et comportement des agents de l'industrie, Colloque "Structures Economiques et Econométrie" - Lyon, 23 - 24 mai 1985.

J.Henkel, 2006, "The Jukebox Mode of Innovation – A Model of Commercial Open Source Development", DRUID Working Paper 06-25.

F. Horn, 2004. L'économie des logiciels, collection repères, la Découverte, 128 p.

M. Iansiti and G. L. Richards, 2006. The Business of Free Software: Enterprise Incentives, Investment, and Motivation in the Open Source Community. Harvard Business School Working Paper Series, No. 07-028.

B. M. Kogut and A. Metiu, 2001. Open Source Software Development and Distributed Innovation, Oxford Review of Economic Policy, Vol. 17, No. 2, Summer 2001.

K. Lakhani and E. von Hippel, 2003. How open source software works: Free user to user assistance. *Research Policy*, 32: 923-943. URL: http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf.

K. Lakhani et R. Wolf, 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, in Feller, J., R. Fitzgerald, S. Hissam, & R. K. Lakhani (Eds.). Perspectives on free and open source software , MIT Press.

L. Muselli, 2002, Licenses: strategic tools for software publishers? In Clément-Fontaine et al., [2002], editor, *New Economic Models, New Software Industry Economy*, pages 129-145.

S. Slatter, 1992. *Gambling on Growth: How to Manage the Small High-Tech Firm*, Wiley, New York.

W.E. Steinmueller, 1996, 'The U.S. Software Industry: An Analysis and Interpretive History', in D. Mowery (ed.), *The international computer software industry: a comparative study of industry evolution and structure*, Oxford University Press:Oxford and NY.

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

J. Tirole (1988) *The Theory of Industrial Organisation*. Cambridge, Mass.: The MIT Press.

E. Von Hippel, 1988, *The Sources of Innovation*, Oxford University Press.

E. Von Hippel, 2002. Open Source Software as horizontal innovation networks – by and for users, MIT Sloan School of Management W.P. N°4366-02.

E. Von Hippel, 2005. Democratizing Innovation, Cambridge, MA: MIT Press (April).

J. West, 2003. How open is open enough? Melding proprietary and open source platform strategies. Research Policy 32 (7), 1259-1285.

J.-B. Zimmermann, 1995, Le concept de grappes technologiques. Un cadre formel. *Revue économique*, 46 (5): 1263-1295.