Competition against Open Source Community in Presence of Forking: Implications for Open Source Policy*

Edmond BARANES† and Mourad ZEROUKHI‡
LAMETA, University of Montpellier 1

April 2011

Abstract

In this paper we study the competition in mixed duopoly in which a proprietary software firm competes against an community of voluntary developers of OS software both in the labor market of programmers and in the product market. Our analysis is based on a two-stage-game, where at the first stage the two entities are engaged in the development of their own software. The development activity is carried out by skilled programmers recruited on a perfectly competitive labor market. The firm pays the same wage to the programmers while the OS community offers them the opportunity to signal their ability, which leads to no pecuniary income. At the second stage, the firm and the OS community distribute their software on the product market. The firm sets the price of its software and the OS community offers its software for free on the Internet. Users evaluate both products and decide to buy the proprietary software or to download the free OS software. The central assumptions are that users and programmers are heterogeneous, and the OS community suffers from forking in software developing. We first describe how the proprietary firm reacts to the competition of the OS community under conditions of users and programmers heterogeneity. Then, we analyze the welfare implications of the public policy supporting OS community.

Key words: open source community, proprietary firm, heterogeneity, forking

*We have benefited from helpful discussions with Eric MALIN, Dominique TORRE, Jean-Christophe POUDOU and Jean-Benoît ZIMMERMANN.
†E-mail: ebaranes@univ-montp1.fr. Phone: +33 (0) 467158317
‡E-mail: mourad.zeroukhi@univ-montp1.fr. Phone: +33 (0) 467158403

1 Introduction

Open Source Software (OSS) has become a serious competitor for the dominant proprietary software (PS). Some successful OSS such as Linux, Apache and Sendmail are challengers (leaders even in the case of Apache) in their respective market segments. For example, the market share of the operating system Linux is around 21.2% behind the traditional leader in the sector, Microsoft Windows with around 73.9% (IDC, 2010)[1]. The Apache web server dominates its market with 54.56%, leaving Microsoft IIS (Internet Information Server) behind with 24.27% (Netcraft, 2010). Other examples including Openoffice.org, MySQL, Firefox and Gimp are emerging as credible alternatives to the proprietary software in their respective markets segments.

These successes of OSS have lead to an increasing interest of the proprietary firms to know about the strategic behaviors to adopt for competing efficiently against OS Community. This interest is best summarized up in the following statement from an internal email written by Steve Ballmer, the Microsoft CEO in 2003: *"Non-commercial software products in general, and Linux in particular, present a competitive challenge for us and for our entire industry, and they require our concentrated focus and attention".*

In this paper we seek to understand how a proprietary firm should compete against an OS community both in the labor market of programmers and in the product markets of users under conditions of preference heterogeneity and in the presence of forking. We analyze also the welfare impact of public's policy supporting OS community.

We investigate these questions using a mixed duopoly model in which a proprietary firm competes against an OS community in a two stages game. At the first stage, the firm and the OS community develop the quality of their software products by hiring programmers on perfectly competitive labor market. The firm pays the same wage to the programmers while the OS community offers them the possibility to signal their abilities, which leads to non-pecuniary rewards (for instance, ego gratification, peer recognition, personnel learning and enjoyment from programming). The quality of the software product depends on the number of programmers working on. At the second stage, the users evaluate the two products and decide whether to buy the PS or to acquire the OSS for free. The users and the programmers are heterogeneous with respect to their preferences for the firm and OS community on their respective markets. Moreover, we assume that the OS community suffers from forking in software developing, which may decrease the quality of its product.

Note that the provider of the OSS (the OS community) is not a strategic player, since there is no price to charge, no wage to pay and no profit to generate. Hence, the price of the PS and the wage fixed by the firm are, in our model, the only strategic variables from the supply side. Note also that the firm has an explicit power to decide on the quality of the OSS by strategically setting its wage.

The main findings of the paper are the following. First, the OSS will not be developed if the level of forking within the OS community is sufficiently high. Second, the proprietary firm needs a critical number of programmers in order to reach a level of quality, which makes its software product more valuable than that of the OS community. Third, depending of the heterogeneity of programmers and users, and on the level of forking, the proprietary firm may reduce the market share of the OS community by strategically setting its wage. Finally, from the welfare perspective, the public policy supporting OS community to reduce its level of forking is socially profitable only if the programmers in the labor market are sufficiently heterogeneous.

This paper is organized as follows. In section 2, we briefly review the related literature on competition between OSS and PS. In section 3 we present the assumptions of the model and the game. Price and wage optimal decisions of firm are analyzed in section 4. In the section 5 we investigate the welfare impact of the public policy supporting the OS community. Section 6 concludes.

[1] IDC, International Data Corporation.

2 Literature

The competition between proprietary firms and OS communities has already been studied in the literature of OS. Bitzer and Schröder (2005) examine the innovation impact of OSS entry in the software market. In their model competition takes place on technology rather than on price or quality. They show that the evolution of the market structure from a monopoly to a duopoly exerts a positive effect on innovation. Lin (2004) proposes a model of duopolistic competition between OSS and PS in which users are heterogeneous in terms of skill and experience. She finds that the market may tip to the OSS if it provides significant benefits to users who can develop customized tools and applications using the OSS. Gaudeul (2004) studies the competition between OSS and PS when the OS community faces imperfect coordination between its programmers. She assumes that users have different valuations of software features and programmers are differentiated in term of programming costs. Gaudeul shows, that in equilibrium, the OSS will always coexist with proprietary software; low-income users and highly skilled programmers opt for an OSS, and all the others adopt a PS. Casadesus-Masanell and Ghemawat (2003) propose a mixed duopoly model to analyze the competition between Linux and Microsoft in a dynamic setting. Linux is characterized by strong demand-side effect and Windows by a larger initial installed base. The authors show that, if Microsoft keeps its initial advantage in terms of installed base and uses its market power to set its price strategically, Linux will never become the market leader. The result of the competition would be the coexistence of two or Linux software exits the market unless the cost difference between the two software increases significantly or strategic buyers such as public institutions and large corporations commit themselves to the development of Linux. Mustonen (2003) addresses the constraints that proprietary software faces, both on the labor market of programmers and on the product market when it competes with an OS community. He assumes that programmers are differentiated in term of productivity and they can choose either to be employed by a proprietary firm or to join the OS community. Users face implementation costs (for instance, the costs of installing and learning the software). Mustonen shows that, if the implementation costs of the OSS are sufficiently low, some users will prefer OSS and the proprietary firm has to take this into account in her pricing strategy. He concludes that the competition of the OSS refrain the proprietary firm from exerting fully its market power, pricing lower and investing more in quality of its product.

Our paper is close to Mustonen (2003) since we also analyze the interactions between a proprietary firm and an OS community in two types of market: labor market and product market. However, our approach differs from his, firstly in that Mustonen (2003) considers the implementation costs of OSS while we consider the effects of forking. Secondly, in Mustonen (2003) the programmers are differentiated in term of productivity, while in our paper they have identical skills and abilities but they are heterogeneous in term of preferences over job characteristics. Finally, we introduce explicitly the spatial dimension of competition between software OS and PS.

3 Model

We consider competition in mixed duopoly in which a proprietary software firm competes against an community of voluntary developers of OS software both in the labor market of programmers and in the product market. Our analysis is based on a two-stage-game, where at the first stage the two entities are engaged in the development of their own software. The development activity is carried out by skilled programmers recruited on a perfectly competitive labor market. The firm pays the same wage to the programmers while the OS community offers them the opportunity to signal their ability which leads to non-monetary rewards. At the second stage, the firm and the OS community distribute their software on the product market. The firm sets the price of its software and the OS community offers its software for free on the Internet. Users evaluate both products and decide to buy the proprietary software or to download the free OS software. The central assumptions are that users and programmers are heterogeneous, and the OS community suffers from forking in software developing. We first describe how the proprietary firm reacts to the competition of the OS community under conditions of users and programmers heterogeneity and in presence of forking. Then, we analyze the welfare implications of the public policy supporting OS community.

3.1 Labor market: Programmers, Firm and OS community

We consider a continuum of N programmers with identical skills and abilities, but they have heterogeneous preferences with respect to software products. We assume that preferences are distributed uniformly along the Hotelling line $[0,1]$. The location of a programmer is denoted $\eta \in [0,1]$ and is associated with his cost of programming software. The parameter $c > 0$ captures the unit coding cost of software or "mismatch" [2]. Here $c\eta$ is a programmer's coding cost of an application for the OSS, and $c(1-\eta)$ is a programmer's coding cost of an application for the PS. Any programmer develops only one application of software per unit of time which correspond to its own productivity and chooses either to work within software firm or to join the OSS community.

Let us suppose that firm pays the same wage, w, to the programmers while the OS community offers them the opportunity to signal their ability[3] which leads to no pecuniary income, r. We define the utility of programmer η, $\eta \in [0,1]$ by

$$U_D \equiv \begin{cases} w - c\eta & \text{when developing a PS} \\ \\ r - c(1-\eta) & \text{when developing a OSS} \end{cases} \tag{1}$$

where $w > 0$ and $r > 0$ represent the gross compensation or utility a programmer would obtain for developing the PS and OSS respectively. We assume that w may be larger or smaller than r. The parameters w and r are assumed to be sufficiently large for the entire labor market to be fully covered (i.e. the labor market is characterized by excess demand, and in principe the two entities would absorb all programmers). Thereby, we preclude monopoly and kink equilibrium and concentrate on competitive one. Notice that this assumption essentially means that all programmers have job, which seems reasonable, at least for the software industry.

Programmers choose either to be employed by the firm or to join the OS community by comparing her net utility from each type of software product. The marginal programmer, denoted by η^*, is indifferent between firm and OS community. The location of this programmer is given by

$$w - c\eta^* = r - c(1-\eta^*) \tag{2}$$

Solving for η^*, we obtain the location of the indifferent programmer to be

$$\eta^* = \frac{1}{2c}(c - r + w) \tag{3}$$

Since programmers are uniformly distributed and those to the left η^* are employed by the firm while those to the right join the OS community, that the number of OS programmers, N_{OSS}, and firm programmers, N_{PS}, are given by

$$N_P = \eta^* N = \frac{N}{2c}(c - r + w) \tag{4}$$

$$N_{OS} = (1-\eta^*)N = \frac{N}{2c}(c + r - w) \tag{5}$$

where $N = N_{PS} + N_{OSS}$ is the size of the labor market.

[2] c may be also interpreted as the disutility from not developing the most preferred software product, the transportation cost or the degree of differentiation in the standard Hotelling model from a programmer's point of view

[3] Lerner and Tirole propose that open source programmers have two incentives to participate: the ego gratification incentive and the career concern incentive. The later is the pure monetary concern and refers to future jobs offers, and shares in commercial open source firms. The ego gratification incentive stems from a desire for peer recognition.

3.2 Product Market: Users, Firm and OS community

Users are uniformly distributed along the Hotelling line $[0,1]$ with respect to their preferences for the software products. The two entities (firm and OS community) are located at the extremes of this Hotelling line, firm is at $x = 0$ and OS community at $x = 1$. The location $x \in [0,1]$ reflects the user's relative preference over the two software products. The user located at $x = 0$ point has a strong preference for the PS, whereas the user located at $x = 1$ has a strong preference for the OSS. Any user adopts only one unit of software and chooses between PS or an OSS. We define the utility of user x, $x \in [0,1]$

$$U_x \equiv \begin{cases} V_{PS} - tx - p & \text{if he buys PS} \\ V_{OSS} - t(1-x) & \text{if he downloads OSS} \end{cases} \tag{6}$$

where $V_{PS} > 0$ and $V_{OSS} > 0$ represent the gross utility (or the maximum quality) of the PS and the OSS respectively, t is the cost of transportation or the disutility from not using the most preferred software product[4], p is the price charged by the firm. The price of OSS is equal to zero by definition. We define $\Delta \equiv V_{PS} - V_{OSS}$. As long $\Delta = 0$ there is no vertical differentiation, while the software products are vertically differentiated when $\Delta \neq 0$.

The quality[5] of a software product depends on the development output of the participating programmers. Proprietary firm determines its software quality by hiring N_{PS} programmers and in doing so, it determines the number of programmers available for joining the OS community N_{OSS}, and thus determines the software quality of the OS community. Programmers have identical skill-level or productivity. For the sake of simplicity, we normalize the programmer marginal productivity to one (unit of software feature per unit of time). The OS community is subject to the risk of forking, i.e. the splitting of the OSS project or its development into a variety of competing applications which may waste resources of the OS community (Lerner and Tirole, 2002). Thus, when forking leads to overlap of programming efforts, the overall quality of OSS may not be improved at the same extend due to a lack of coordination and support. Note that there is no risk of forking in the case of PS because the proprietary firms encourage and facilitate close cooperation between programmers. Therefore, the effective overall software qualities for respectively the firm and the OS community are defined by the following production functions:

$$V_{PS} = (1-\lambda)\, N_{PS} = \frac{N(1-\lambda)}{2c}(c - r + w) \tag{7}$$

$$V_{OSS} = (1-\lambda)\, N_{OSS} = \frac{N(1-\lambda)}{2c}(c + r - w) \tag{8}$$

In this modelisation, λ is the parameter of forking normalised to 1 for the firm, N_{PS} and N_{OSS} represent the number of programmers developing the PS and OSS respectively. Note that the parameter λ will take a value between 0 and 1 in the OS community. This because the OSS projects might be less coordinated (Mishra et al., 2005).

The firm hire programmers at a wage of w and the marginal costs of production are zero because software is information good (*i.e.* easily reproducible). The total development cost to the firm, $C(N_{PS})$, is simply the sum of wage hiring programmers:

$$C(N_{PS}) = wN_{PS} = \frac{wN}{2c}(c - r + w) \tag{9}$$

Note that the OSS is developed at zero fixed cost because the OS programmers work for free (i.e. they are volunteers) and receive non monetary rewards, r.

[4] The parameter t may be interpreted in differente ways: the cost of learning how to use the the software, the installation cost or the degree of differentiation in the standard Hotelling model from a user's point of view

[5] Quality of software means two things: first software that has fewer bugs is of higher quality. Second, software with more features (i.e. graphical analysis, regression analysis etc. in mathematical analysis software) is of a higher quality. All users value the software equally at its quality.

4 Equilibrium Analysis

In our model, a market equilibrium requires solving a two-stage game involving, first a wage-setting problem whose solution is referred to as a labor market equilibrium and, then, a price-setting problem whose solution is described by a product market equilibrium. In other words, market equilibrium is a sub-game perfect Nash equilibrium. As result, the model is solved by backward induction.

4.1 Product-Market Equilibrium

Assuming that both entities (firm and OS community) have a positive demand there is one user located at $\overline{x}$ who is just indifferent between adopting PS or OSS, such that

$$V_{PS} - t\overline{x} - p = V_{OSS} - t(1 - \overline{x}) \tag{10}$$

This user is given by

$$\overline{x} = \frac{V_{PS} - V_{OSS} - p + t}{2t} \tag{11}$$

All users to the left of $\overline{x}$ buy the PS; all users to the right of $\overline{x}$ download the OSS. The market shares for the firm and the OS community are respectively given by $d_{PS} = \overline{x}$ and $d_{OSS} = 1 - \overline{x}$.

Thus, the firm's profit is given by

$$\Pi = p\left(\frac{V_{PS} - V_{OSS} - p + t}{2t}\right) - wN_{PS} \tag{12}$$

and the profit of OS community is equal to zero by definition. The Firm knows the quality levels (V_{PS}, V_{OSS}) and chooses its price in order to maximize profit. The firm's profit maximizing yields the optimal price

$$p = \frac{t + V_{PS} - V_{OSS}}{2}, \tag{13}$$

and then the optimal demand

$$\overline{x} = \frac{1}{4} + \frac{(V_{PS} - V_{OSS})}{4t} \tag{14}$$

and the firm's profit

$$\Pi = \frac{(t + V_{PS} - V_{OS})^2}{8t} - wN_{PS} \tag{15}$$

Note that the market sustains two software products if and only if

$$t > \frac{|V_{PS} - V_{OSS}|}{3} \tag{16}$$

If this condition is violated the quality difference between the two software is so large, that only the software with the superior quality will survive on the market. Note also that this condition implies that the firm's profit is strictly increasing in t.

From (11) we can see that if the quality of the OSS is equal to that of the PS, $V_{PS} = V_{OS}$, the OS community gets a strictly larger market share than the proprietary firm. This because the PS software is sold at a positive price, while the OSS is distributed for free. Note that if the firm sets its price to zero and $V_{PS} = V_{OS}$, the firm and the OS community will get the same market share. However, in this case firm's profit is also zero. Finally, the firm can set a positive price, $p > 0$ and gets a larger market share if the quality of its software is higher than that of the OS software, $V_{PS} = V_{OS}$. In this model, we have chosen to focus on the most meaningful cases in which the profit is positive.

The following proposition establishes the conditions under which the firm may produce software of identical or higher quality than the software provided by the OS community.

Proposition 1 *Under the condition (16), there existe a minimum wage level,* $\underline{w} > 0$*, which depends on the parameters* r*,* c *and* λ*, as such as for* $w \geq \underline{w}$*, the quality of the PS is higher or equal than the quality of the OSS,* $V_{PS} \geq V_{OSS}$*.*
Proof. *See the Appendix* ■

Accordingly to Proposition 1, the quality of PS depends on what wage the firm has to pay programmers. If the firm chooses to develop the same quality software that the OSS, it leaves a large market share for the OS community, by setting its wage equal to the minimum wage, $w = \underline{w}$. However, if the firm chooses to develop higher software quality than that of the OSS, the wage it pays must exceed the minimum wage, $w \geq \underline{w}$ and gets a larger market share. Using (7) and (8) the condition can be stylized as:

$$V_{PS} \geq V_{OSS} \Rightarrow w \geq \frac{(2r - \lambda (c + r))}{(2 - \lambda)} = \underline{w} \tag{17}$$

4.2 Labor-Market Equilibrium

In this section, we study the first stage of the model in which firm set wage, anticipating the resulting equilibrium in the product-market.

By substituting (4), (7) and (8) into (15), we obtain the profit function of the firm at the first stage of the game, i.e., at the development stage. This profit function is given by

$$\Pi = \frac{1}{8t} \left(t + \frac{N \left(2(w - r) + \lambda(c + r - w)\right)}{2c} \right)^2 - \frac{wN}{2c} (c - r + w) \tag{18}$$

The firm choose w to maximize (18) subject to (16) and (17). The proposition (2) describes in details this maximization (i.e., the firm's optimal choice of w)

Proposition 2 *When*

$$t > \frac{N \left(c(3\lambda - 2) + (\lambda - 2)(r + \lambda - 2)\right)}{12c} = \underline{t} \tag{19}$$

and

$$c < \frac{\left(-8r + 4r\lambda + (\lambda - 2)^2\right)}{4(2 - 3\lambda)} = \underline{c} \tag{20}$$

there exists a unique symmetric market equilibrium in which the firm wage is equal to

$$w^* = \frac{N (2 - \lambda)(c\lambda - 2r + r\lambda) - 2ct (4c - 4r + \lambda - 2)}{16ct - N (\lambda - 2)^2} \tag{21}$$

and the price(p^**), demand (* d^*_{PS}*) and the number of programmers hired by the firm (* N^*_{PS}*) are respectively given by* (13) *and* (16) *with* (21) *substituted in place of* w*.*

Proof. See the Appendix. ■

The condition (19), which is sufficient for the coexistence of the proprietary firm and the OS community, holds when t is large enough and/or the parameters c, r and λ are low enough. The condition (20), which guarantees that the maximum quality of the PS is the same or higher than the maximum quality of the OSS, i.e. $\Delta (w = w^*) \geq 0$, holds when c is low enough and/or the parameters r and λ are low enough. The wage (21) equals the revenue/utility of the OSS programmers ($w^* = r$) when all the cost parameters c and t are zero, namely, when there are no labor and product market heterogeneities. In addition, the wage (21) equals the minimum wage ($w^* = \underline{w}$) when ($c = \underline{c}$).

It is now possible to determine how changes in the parameters λ, N and r affects wage, price, demand, labor supply and profit of the firm. These effects are described in a set of corollaries below.

Corollary 1 *When the level of differentiation between software products with regards to user preferences* ($t > \underline{t}$) *is higher than the level of differentiation with regards to programmer preferences* ($c < \underline{c}$)*, the wage of firm (w) increases for a higher level of forking* (λ)*, for a larger labor market* (N)*, and for a higher OS programmers' revenue* (r)

Proof. See the Appendix. ■

The most surprising result in Corollary 1 is that the wage of firm can actually increase as the level of forking increases (i.e. as the level of coordination among OS programmers decreases). This result is counterintuitive since the firm does not take advantage from the declining quality of the OSS to reduce its development cost by lowering its wage in the labor market. The reason is that, to maximize its profit, the firm considers the positive impact of higher wage in the labor market on its demand in the product market. Clearly, if the firm chooses to reduce its wage when λ increases, it must set a lower price compared to the equilibrium price to increase its demand. Since competition in the labor market is more intense than that in the product market ($c < \underline{c}$, $t > \underline{t}$), the decreases in its demand due to lower wage is sufficiently large that it outweigh the positive lower price effect on its demand and thus the profit decreases. The same argument is used to explain the positive correlation between the equilibrium wage of and the size of labor market. Finally, it's intuitive that the equilibrium wage increases in OS programmers' revenue.

Corollary 2 *When the level of differentiation between software products with regards to user preferences* ($t > \underline{t}$) *is higher than the level of differentiation with regards to programmer preferences* ($c < \underline{c}$)*, the equilibrium price, demand and labor supply of firm are increasing for a higher level of forking* (λ)*, for a higher size of labor market* (N)*, and for a higher OS programmers' revenue* (r)*.*

Proof. See the Appendix. ■

An increase in λ increases the number of programmers hired by the firm because the firm has stronger incentives to compete with OS community on the basis of quality (because its initial market share is higher). Thus, an increase in quality of the PS triggers a positive feedback effect on firm's price and market share by making its software product relatively more valuable in the eyes of users than that of OS community. The increase in the size of labor market plays similarly. Corollary 2 shows also that increasing in the OS programmers' revenue has a negative effect on the demand, price and labor supply of firm. The intuition of this result is as follows. A marginal increase in the revenue of OS programmers causes PS quality to drop due to the decrease in the number of programmers working on. In this context, it's optimal for the firm to reduce development cost by reducing the number of hired programmers and try to increase its demand by reducing price rather than competing aggressively with the OS community on the basis of quality.

Since the price, demand and labor supply are positively correlated as shown in Corollary 2, one might conclude that the comparative statics of firm's profit is straighforward. However, this is not the case because firm's profit is also affected by the development cost. indeed, corollary below shows that there are some surprises.

Corollary 3 *When the level of differentiation between software products with regards to user preferences* ($t > \underline{t}$) *is higher than the level of differentiation with regards to programmer preferences* ($c < \underline{c}$)*, the profit of firm is increasing in the level of forking* (λ)*, but decreasing in the size of labor market* (N) *and the OS programmers' revenue* (r)*.*

Proof. See the Appendix. ■

The most surprising finding of above Corollary is that the profit of firm can decrease with N. At first sight this result is counterintuitive because the "duopsony" power of the firm (measured by the wage it pays for programmers) decreases rather then increases with the size of labor market. The raison is that the wage paid by the firm remains high despite the increased size of the labor market because the heterogeneity of programmers is low enough ($c < \underline{c}$). Therefore, the firm's benefit due to the increased quality (because it hire more programmers when N increases) is insufficient to offset the development cost and thus the profit of firm decreases

5 Welfare and policy implications

In this section, we analyze the implications of public policy supporting OS community on welfare, focusing on the possibility that helping OS community to reduce its level of forking (i.e. to better coordinate the development efforts of its programmers) can induce the socially optimal level of the quality software. Such a policy may lead users to partially internalize this externality and take it into account when choosing the two software products. As a result, the users' reservation price for the PS decreases.

The public policy supporting OS community can take the form of subsidizing institution of the OS movement that try to coordinate software development and standard setting (Schmidt and Schnitzer, 2003)[6]. This policy may encourage and facilitates close collaboration between OS programmers and limits the treat of forking in the OS community. We can model this policy as an decrease of λ. Notice that the decrease in λ can also be the result of maturity of the OS community itself, i.e. it becomes more effective to coordinate the effors of its volunteers programmeurs. To clarify the analysis and facilitate interpretation of the results, let's suppose that the intervention of the public authority makes it possible to reduce the level of forking to zero. Thus two cases are distinguished: the case before the public intervention, i.e. the case in which $\lambda > 0$ and the case after the public intervention, where $\lambda = 0$.

As can be seen from Proposition 2, when the heterogeneity of programmers in the labor market reaches the threshold level $c = \frac{(\lambda+1)(1+\lambda-4r)}{12\lambda-4} = \underline{c}$, the proprietary firm under-invest in the quality of its product by setting its wage equal to the minimum wage ($w^* = \underline{w}$) so that ($V_{PS} = V_{OSS}$). In the analysis that follows, we compare social welfare for that particular case before and after public intervention.

In this model, we consider that the price paid by users and the wage earned by programmers as a pure transfer that does not affect social welfare. This is due to the single-homing nature of our Hotelling model. Thus, we measure the welfare as the net of surplus, that is the sum of firm's profit and users' surplus minus the disutility of programmers. Formally, the function of total welfare can be written:

$$W = \int_0^x (V_P - tx)\,dx + \int_x^1 (V_{OS} - t(1-x))\,dx - \int_0^\eta (c\eta)\,d\eta - \int_\eta^1 (c(1-\eta))\,d\eta - wN_P \tag{22}$$

Before the public intervention and when $c = \underline{c}$, total welfare equals:

$$W(\lambda > 0)\big|_{c=\underline{c}} = \frac{\left(24 - 16r - 20t - 60\lambda + 48r\lambda + 40t\lambda + 28\lambda^2 + 6\lambda^3 - 24r\lambda^2 - 15t\lambda^2\right)}{48\lambda^2 - 128\lambda + 64} \tag{23}$$

After the public intervention and when when $c = \underline{c}$, total welfare is

$$W = W(\lambda = 0)|_{c=\underline{c}} = \frac{6 - 5t - 4r}{16} \tag{24}$$

We define $\Delta W|_{c=\underline{c}} \equiv W(\lambda = 0)|_{c=\underline{c}} - W(\lambda > 0)\big|_{c=\underline{c}}$

The results of comparison between social welfare before and after public intervention are summarized in the next proposition.

Proposition 3 *When the level of differentiation with regards to user preferences is high ($t > \underline{t}$) and the level of differentiation with regards to programmer preferences reaches the threshold level ($c = \underline{c}$), helping OS community to reduce its level of forking, increases welfare.*

Proof. From (1) and (2) its follows that ∎

$$\Delta W|_{c=\underline{c}} = \frac{\lambda\left(6 - 8r - 5\lambda + 6r\lambda - 3\lambda^2\right)}{8\left(4 - 8\lambda + 3\lambda^2\right)} > 0 \ \ \forall\lambda \left]0, \frac{2}{3}\right[\text{ and } r < \frac{(2-\lambda)}{4}$$

[6] *An example is BerliOS, a mediator for open source developers and customers, which is co-funded by the German federal government and private companies such as Hewlett-Packard and Linux Information Systems*

Intuitively, helping OS community to better coordinate the development efforts of its volunteer's programmers (i.e. to reduce its level of forking) increases the quality of its software product, decreases the profit of the proprietary firm, increases the surplus of users, and decreases the disutility of programmers. Since the market share of the proprietary firm is strictly lower than that of the OS community when c reaches the threshold value $\underline{c}$, the decreases in its profit is not sufficiently large to outweigh the all others positive effects and thus the total welfare increases.

6 Conclusion

Open source software has become a serious competitor of proprietary software not only in product market but also in the labor market. In the labor market, the OS community (the provider of OSS) creates an alternative job career for the programmers, while in the product market it offers an opportunity for users to download the OSS for free as an alternative to PS. Facing this competition, proprietary firms have begun to consider how to deal with the presence of the OS community. In this paper, we propose a theoretical model to study how a proprietary firm interacts with an OS community both in the labor market and in the product market under conditions of preferences heterogeneity and when the OSS development suffers from forking. In particular, we focus on the optimal price and quality of a proprietary firm. The quality is endogenous and is dependent on the number of programmers developing the PS and the firm can determine this number by paying wage. We study also the impact on welfare of public policy supporting OS community to reduce its level of forking.

Our study has relevant implication for the firm behavior. It suggests that if the level of differentiation between OSS and PS from the users' perspective is high and that from the programmers' perspective is low, the firm will invest more in quality of its product by setting higher wage in the labor market to make it unprofitable for the programmers to join the OS community. Conversely, if the level of differentiation between OSS and PS from the programmers point of view is quite high, but less than that from the users point of view, the firm will not engage in an aggressive quality strategy and leave a large market share for the OS community.

Our study has also implications for public policy. It suggests that if the government can correctly evaluate the magnitude of programmers heterogeneity in the labor market, it can increase the quality of software available to users in the product market by helping the OS community to better coordinate the development efforts of its volunteers programmers and consequently decrease its level of forking. Thus, finding that programmers are relatively more heterogeneous in the labor market should be an argument for this public intervention.

Our results however, are subject to the limitations of the model. First, we assume that all programmers are identical skills and abilities. This for obvious raisons is not realistic. A direction of future research could be to model heterogeneity in skills and abilities. Second, in this paper we do not consider the opportunity for programmers to multi-home. Future research can look at multi-home as strategic choice by programmers. Finally, another direction of future research will be to consider the Network externalities in software market.

References

[1] Gaudeul, A., (2004), "Competition between open source and proprietary software: the (La) TeX case study", working paper, http://agaudeul.free.fr.

[2] Gaudeul, A., (2005), "Competition between open source and proprietary organization", working paper, university of East Anglia - Norwish and ESRC Centre for Competition Policy.

[3] Baake P.,.Wichmann, T., (2003), "Open source software, competition and potential entry", Berlecon Research papers 005.

[4] Bessen, J., (2004), "Open source software: Free Provision of Complex Public Goods", http://ssrn.com/abstract=588

[5] Bonaccorsi, A., C. Rossi, (2003), "Why open source software can succeed, Research Policy, Special Issue on open source software development", 32(7), 1243-1258.

[6] Browne C., (1999), "The economics of free software", www.ntlug.org/~cbbrowne /freecon.html.

[7] Casadesus-Masanell, R., and P., Ghemawat, (2003), "Dynamic Mixed Duopoly : A Model Motivated by Linux vs. Windows", Strategy Unit Working Paper 04-012, graduate School of Business Administration, Harvard University.

[8] Dalle J-M., Juillien N (1999) Nt vs. Linux, or some explorations into the economics of free software, mimeo, ENS Cachan & IDHE-CNRS.

[9] Dasgupta P., David P. (1994) Toward a new economics in science, Research Policy 23(5), 487-521.

[10] Franke, N., E., von Hippel, (2003), "Satisfaying heterogeneous user needs via innovation toolkits: the case of Apache security software", Research Policy, Special Issue on Open Source Software, 32 (7), 1199-1215.

[11] Johnson, J.P., (2002), "Open Source Software: Private provision of public good", Journal of Economics and Management Strategy, 11 (4), pp. 637- 662.

[12] Johnson, J., (2004), "Collaboration, Peer Review and Open Source Software", Working Paper, http://ssrn.com/absract=535022.

[13] Schmidt K., Schnitzer M., (2003), " public subsidies for open source? Some economic policy issues of the software market", CERP Discussion paper.

[14] Kuan J., (2001), "open source software as consummer integration into production", Haas School of Business, University of Califirnia- Berkeley.

[15] Lerner J., Tirole J., (2004), "The Economics of Technology Sharing: Open Source and Beyond", working paper 10956.

[16] Lerner J., Tirole J., (2002), "The Simple Economics of Open Source", The Journal of industrial economics volume L 0022-1821. No.2.

[16] Lerner J., Tirole J., (2001), "The Open Source Movement: Key Reseach Questions". European Economic Rewiew 45, 819-826.

[17] Bitzer.J, (2004), " commercial versus open source software: the role of product heterogeneity in competition", Economic systems 28, 369-381.

[18] Bitzer J., Schröder P., (2005), "the impact of entry and competition by open source software on innovation activity", Working paper, Free university Berlin Germany and Aarhus School of Business, Denmark.

[19] Bitzer J., Schröder P, (2005), "bug- fixing and code- writing: the private provision of open source software". Information Economics and Policy 17, 389–406.

[20] Lihui H., (2004), "Competition between Proprietary and Open Source Software with Heterogeneous Users", Working paper, Boston University School of Management.

[21] Mustonen M., (2003), " copyleft- the economics of Linux and other open source software", Information Economics and Policy, 15 (1), pp. 99- 121.

[22] Raymond, E.S, Cathedral & the Bazaar: Measuring on Linux and Open Source by an Accidental Revolutionary. O'Reilly, Cambridge, MA.