



Mikko Leppämäki – Mikko Mustonen

SIGNALING AND SCREENING
WITH OPEN SOURCE PROGRAMMING

Mikko Leppämäki – Mikko Mustonen

SIGNALING AND SCREENING
WITH OPEN SOURCE PROGRAMMING

Finance

September
2004

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
PL 1210
FIN-00101 HELSINKI
FINLAND

© Mikko Leppämäki, Mikko Mustonen and
Helsinki School of Economics

ISSN 1795-1828
ISBN 951-791-881-X (Electronic working paper)

Helsinki School of Economics -
HeSE print 2004

Signaling and Screening with Open Source Programming¹

Mikko Leppämäki²
Helsinki School of Economics, RUESG and HECER
&
Mikko Mustonen³
Helsinki School of Economics and HECER

September 13, 2004

¹We thank seminar and workshop participants at RUESG, NORIO 2003, EARIE 2003, EALE 2003, and University of Oulu for their comments. Especially we would like to thank Mathias Dewatripont, Neil Gandal and Jorgen Weibull for useful comments and suggestions.

²**Address:** Department of Accounting and Finance, Helsinki School of Economics, P.O. Box 1210, FI-00101 Helsinki, Finland. e-mail: mikko.leppamaki@hkkk.fi

³**Address:** Department of Economics, P.O. Box 54, FIN-00014 University of Helsinki, Finland. e-mail: mikko.mustonen@helsinki.fi

Abstract

In our paper, individual programmers participate in OS programming to signal their programming skills to commercial software companies and companies use OS projects as a screening device for hiring new workers. The key feature of the model is that the signaling activity itself creates a market externality affecting the commercial software companies' profits as the signaling activity creates a program that will be freely available to consumers. We show that in the least cost separating equilibrium, the programming credential in the case of a free substitute (complement) OS software is lower (higher) than in the case of independent programs. We also show that for the software companies' management, OS projects may function as clever screening mechanisms for new workers without committing to hire anyone before they have actually shown their capability as a programmer. Finally, there is a conflict of interest between the private and social incentives over the choice of an OS project when the size of the market for consumer software is small; the society would prefer a substitute OS project rather than complement.

JEL Classification: D23, D82

Keywords: signalling, screening, professional labor market, programming, open source software, OSS

1 Introduction

This paper is inspired by two observations from the open source software environment and software industry in general. It is widely documented that individual programmers develop software in open source environments that provide no immediate direct (monetary) payoffs for their programming effort⁴. Why do they get involved in such ventures? Equally well, commercial software companies do support and subsidize such open source communities in various ways.⁵ What is the motivation behind firms' behavior given that these communities may potentially be even harmful by providing free open source programs that compete with their own commercial (copyright) programs?

We show within the formal yet simple model that indeed such behavior is fully rational. On one hand it can be explained by the incentives of independent programmers to participate in open source projects in order to signal their innate programming ability to commercial software companies. On the other hand, from the software companies' management point of view, the open source projects function as screening mechanisms for hiring new competent and highly talented workers. The financial support for OS projects can thus be seen as a clever managerial practice of motivating potential employees to demonstrate their programming capability without committing to hire anyone at that stage. It is only after someone has shown his talent in OS programming by getting a high status ("programming credential") within the OS project that the software company is willing to hire and pay a high salary to a top performer.

We base our analysis on a skill signaling approach in OS environments (c.f. for instance Lerner and Tirole 2002) and extend the signaling model first presented by Spence (1973, 1974, 2002) into a case where the signalling activity in the programmers' professional labour market creates an external-

⁴For descriptions of open source, see for instance Raymond 1998, Browne 1999, GNU 2000a, 2000b, Kogut and Metiu 2001 and Lerner and Tirole 2002.

⁵See for instance IBM News 2001, Collab.net 2001, Hann et al. 2002, Mustonen 2004.

ity affecting the software companies' profits.⁶ We assume that the signaling activity itself creates as a by-product a program that will be freely available to consumers, and the externality effect arises via the interaction of the freely available OS program and commercial software. Our modelling choices allow us to cover both positive and negative market externalities. That is, we consider both complementary and substitute OS programs. The motivating examples from software market that we have in our minds are for instance Openoffice that is likely to be a complement to commercial operating systems like Windows, whereas Linux is a substitute for them. Interestingly, in our case the signalling activity as such is not social waste since it will eventually realize in a free software that consumers value, and clearly this is valuable for the society as well.

Our model has three types of players; programmers, software companies and consumers, who will interact in two types of markets: the professional labor market and the software market. We start by examining programmers' (who differ in their innate ability) incentives in participating in OS projects and software companies' hiring and derive a Perfect Bayesian Equilibrium (PBE) of the signaling game. In particular, we show that the programming credential that the high productivity programmer needs to attain in order to separate from the low productivity programmer in the case of free substitute OS software is lower than in the case of independent programs. This is due to the fact that in the case of substitutes, the software company's profits are reduced due to a negative market externality ("intensified competition"), which will also reduce wages, and thus in the equilibrium the least cost separating programming credential is reduced as well. This implies that in terms of "quality" (as lines of code transforming into the functionality of a program), the substitute open source program has lower quality than the independent program.

Similarly, we show that in the case of a complementary program due to a

⁶In our companion paper Leppämäki and Mustonen (2004) we consider the signaling with positive and negative externalities in more general framework and provide a complete characterization of the equilibria with respect to the magnitude of externalities and market size.

positive market externality ("increased willingness to pay for the commercial software") the profits of the software company are now higher, which will be reflected in higher wages, and therefore the programming credential in the least cost equilibrium will be higher than in the case of independent programs. Thus the resulting free complementary open source programs will have higher "quality".

We also examine the role of an open source project as a possible screening device from the software companies' point of view. We assume there exists a high enough outside option (wage) that implies that the commercial software companies cannot even exist without screening of workers. We show that commercial software companies may indeed be willing to subsidize open source projects in order to be able to screen workers to enjoy non-negative profits.

Finally, we analyze private and social incentives concerning the choice of an OS project. Our main analysis reveals that an individual programmer has an incentive to devote his attention towards OS projects where the outcome of the signalling activity has some complementarities with the commercially produced software. However, this need not always be in the interests of society, and thus we derive the exact conditions under which the society as such would prefer OS projects that will materialize in free substitute programs rather than complementary ones. This will depend most importantly on the size of the consumer market. When the size of the market is small, the society would prefer OS projects that will materialize in substitute programs. And when the consumer market for software is large, then both the individual programmers and society would prefer complementary OS projects.

It is also interesting and useful to contrast our theoretical model and results with the practices of the software industry. There is rather strong empirical evidence that software companies do hire new workers among the top performers in various OS projects. In reality, open source software projects often have home pages on internet where they post merit-based ranking lists of the most important contributors (Hann et al. 2002, Linux-PAM 2003). They could even be called as "hall of fames". The fact that a

programmer is at the top of the list implies that he/she has contributed to the project in a major way by providing important or even crucial pieces of code. We interpret this as being a *programming credential* (in our theoretical model) that signals the programmers' innate abilities. That is, for a more able programmer it is easier to get the top position at the merit based list.

In their recent paper Hann, Roberts and Slaughter (2002) provide first empirical evidence of economic incentives of individual programmers within the Apache web-server open source project.⁷ Their empirical results confirm the existence of economic returns; participation per se as measured by the numbers of contributions made does not lead to wage increases, but a higher status in a merit-based ranking does lead to significantly higher wages. A higher status in a merit-based ranking list is a credible signal of the productive capacity of a programmer. Thus their work gives support for the delayed returns argument - motivation for participation is skill signalling⁸. Commercial software companies are willing to pay for high wages for the top performers i.e. they interpret a high position at the merit-based ranking list being a credible signal of high innate productivity. One could even interpret this type of hiring practices within OS communities as outsourcing of personnel management and recruiting activities of commercial software companies. In the recent empirical paper by Fershtman and Gandal (2004) the authors found that skill signaling is the main motive for programming effort in OS projects that employ a restrictive (eg. GPL) license.

Equally well, it is public information that commercial software companies do support OS projects in various of ways. It is important to notice that the support need not be in the form of direct funding. In fact, the support can take any form that simply helps the formation of OS projects. For instance, over 100.000 independent OS programmers have registered to IBM's services to offer their programs just to Linux open source projects related to IBM. In the relationship, the programmers and IBM exchange information. (IBM 2004)

⁷Apache is used in 63% of the world's over 100 million web servers (Netcraft 2001).

⁸For discussions on delayed returns, we refer to Dasgupta and David (1987,1994), Dewatripont, Jewitt and Tirole (1999) and Stern (1999).

In the relatively new open source economics literature one can distinguish two lines of research; the incentive based approach (e.g. Bessen 2001, Johnson 2002 and Bitzer and Schröder 2002) and consumer market analysis (see Lerner and Tirole 2002, Mustonen 2003, 2004)⁹ In this paper we combine the two lines of research and propose a single model that incorporates the idea of signalling in professional labor market that creates an externality effect in software market due to appearance of free open source software, and where the externality effect may either be harmful or beneficial to the receiver of signal. The novelty of this paper is that it introduces a new way for the signalling to work and it also provides a new motivation for firms to participate in and subsidize open source projects.

The structure of the rest of the paper is as follows. In next section we present the model and the main analysis is carried out in section 3. In section 4 we show that commercial software companies do have incentives to support open source projects in order to screen their work force. Section 5 focuses on the individual programmers' and society's incentives over the choice of an OS projects while section 6 concludes.

2 The Model

In this section we set up the model with three players who will interact in two markets. We consider *programmers* who develop computer software either within the OS projects or within *commercial software companies*. There are also *consumers* who value available computer software. The two markets we consider are *professional labour markets* where computer programmers and software companies interact and computer *software market* where consumers decide whether they buy a commercially produced software or acquire an open source software for free.

⁹See also Schiff (2002) who provides a survey of the early literature of OSS.

2.1 Professional Labour Market for Programmers

We assume that there exists two types of programmers, who differ only in their innate non-verifiable programming ability. It is assumed that the high productivity programmers, so called "good" type have a productivity of $\theta = 2$ and the low productivity programmers, so called "bad" type have a productivity of $\theta = 1$, and that the share of "good" types is equal to q .

In order to separate from the low productivity programmer, the high productivity programmer may engage himself in an open source (OS) programming project. If an individual programmer's contribution to the OS programming project is large enough, his name will appear in the project's 'top contributors' or 'hall of fame' list, which is public information.¹⁰ We call this a *programming credential* and label it by y , $0 \leq y \leq y^{\max}$. Notice that y bears a similarity to Spence's term 'education level' in the sense that a given credential is more easily attainable for a high productivity programmer. Interestingly, within the context of OS programming, the programming credential may have an externality effect. The programmer's effort to separate (i.e. to signal his ability) is directed to writing program code to the OS program. The utility of the programmer is assumed to depend on the wage and the disutility of attaining the credential:

$$U_G = w - \frac{y}{2}, U_B = w - y.$$

In above G refers to "good" type and B to "bad" type, each attaining credential y and earning the wage w . We extend the original signaling analysis of Spence by focusing on the case where the activity of attaining the programming credential y creates as a by-product an OS program that will be freely available for consumers. Thus it will interact with the commercially produced program in the consumer software market.

2.2 Software Market

In our analysis, the software company is a profit-maximising monopoly supplying the commercial program in the consumer market niche. The monop-

¹⁰In reality such information is often posted on the open source projects' homepages in the internet. See Linux -PAM (2003) and consult Hann et al (2002).

olist employs a programmer with productivity θ , and creates the commercial program in a project of size one. We assume that a program's functionality and thus quality in the eyes of the consumer is proportional to the amount of code it contains. The amount of code the commercial program contains and thus its quality¹¹ is assumed to be equal to θ .

We define that for a programming credential level of y , the programmer's effort is manifested in $|k|y$ lines of program code that consumers value. The signalling activity, ie. the effort to acquire a credential y , creates thus a free open source (OS) program of quality $|k|y$ in the consumer market. We distinguish between three scenarios. We define that for $k = 0$, the free OS program is independent of the monopolist's market, for $k > 0$, it is a complement to the monopolist's program, and for $k < 0$, the OS program is an incomplete substitute to the monopolist's commercial good.

At the outset we assume that there exist $4M$ consumers who differ in their willingness to pay for the software, and where M is the measure of market size. When only the commercial firm's software is available ($k = 0$), consumers' valuation of it is evenly distributed on the interval $[0, \theta]$. If the signaling activity has resulted in a complement software to the firm's commercial program ($k > 0$), consumers' valuation of it is increased and are now on the interval $[0, \theta + ky]$. Finally, if the signaling activity produces as by-product a free substitute software ($k < 0$), consumers valuations of the commercial software are as before on the interval $[0, \theta]$ and valuations of the free software on $[0, -ky]$. We assume throughout the analysis that the ratio of valuations is equal for all consumers.¹²

In the case of independent signaling activity, the marginal consumer's

¹¹Just as Fershtman and Gandal (2004), we measure the programmer's contribution in terms of lines of program code. Acknowledging the heterogeneity of the programming languages, we refer to their finding that a strong majority of OS projects are written in C or C++ . Furthermore, such languages are used in a standardized fashion. This allows us to project the contribution to functions of the program which in turn determine its value to consumers.

¹²The (consumer) market analysis draws on Kobolt (1995). Recently it has been elaborated and used to model interaction between commercial and free open source (OS) goods in the software industry by Mustonen (2003, 2004).

net valuation of the commercial firm's software is zero, $V_m - p = 0$, where V_m is the marginal consumer's valuation of the firm's commercial software and p refers to price. The distribution of willingness to pay of consumers implies that the number of consumers that have a higher willingness to pay than V_m is $\frac{\theta - V_m}{\theta} 4M$. Maximization of profit function $p \left(\frac{\theta - p}{\theta} 4M \right)$ yields the optimal price $\frac{\theta}{2}$ and output $2M$. If the signaling activity creates as a by product the software that is a complement, the consumers' valuations are increased compared to the previous case. The number of consumers with a willingness to pay higher than V_m is now equal to $\frac{(\theta + ky) - V_m}{\theta + ky} 4M$. Analogously, the optimal price is $\frac{\theta + ky}{2}$ and output $2M$.

The signaling activity may also create a substitute to the commercial software i.e. $k < 0$. Now the software company has to take into account the competing freely available OS software when pricing it's own program. The surplus of the marginal consumer between the commercial software and the OS software has to be equal, $V_m - p = V_{OS} - 0$, where V_{OS} is the marginal consumer's valuation of the OS software. We know that $\frac{V_m}{V_{OS}} = \frac{\theta}{-ky}$, and developing the marginal condition yields $V_m = p \frac{\theta}{\theta + ky}$. As in the case of independent signaling activity, the number of consumers with higher willingness to pay for the firm's good is $\frac{\theta - V_m}{\theta} 4M$. Inserting the marginal consumer's valuation, profit maximization yields again the optimal price $\frac{\theta + ky}{2}$ and output $2M$.

Thus we can cover all three cases by using the same revenue function, and letting just k to vary

$$R = M(\theta + ky),$$

where M captures the "market size effect", and where k may either be positive, negative or zero. Note that though the revenue functions are identical in the cases of complement and substitute OS programs, the market outcomes are quite different. For a complement, the same consumers that would have bought the commercial program anyway are willing to pay more of that very program. In the case of substitutes some consumers buy the commercial program, and the rest of the consumers acquire the competing freely available OS program.

Following the literature, we assume that the programmer has all the bargaining power in the professional labor market, which implies that the firms will compete for the programmers, and end up with zero profits

$$\pi = R - w = M(\theta + ky) - w = 0.$$

Timing of the model is standard. At the outset nature assigns programmers' productivities and the proportion of high and low productivity programmers. Then a programmer may engage himself in obtaining a programming credential y , and thus simultaneously creating a free OS program that is either a complement or substitute to the commercially produced one. A firm hires a programmer with a programming credential equal to y and pays out wage w . In the case of substitute programs, consumers buy the commercial program or acquire the free good, and in the case of complementary programs, the consumers are willing to pay more for the commercial program. At the end profits are realized and wages are paid out.

3 The Analysis

In the professional labor market, strategies (y_B^*, y_G^*, w^*) and a system of beliefs β^* form the Perfect Bayesian Equilibrium (PBE). To build an equilibrium it is assumed that the attained programming credential is regarded by commercial software companies as a credible signal of a programmer's innate ability. We assume that each programmer chooses the level of programming credential given the wage function $w^*(\theta, y) = M(\theta + ky)$. The "bad" type thus faces a problem

$$y_B^* \in \arg \max_y [w^*(1, y) - y],$$

and the "good" type

$$y_G^* \in \arg \max_y [w^*(2, y) - \frac{y}{2}].$$

The software company hires a programmer with a credential y at wage

$$w^*(\theta, y) = \beta^*(1 | y)M(1 + ky_B^*) + (1 - \beta^*(1 | y))M(2 + ky_G^*) \quad (1)$$

with beliefs β^* that are consistent with equilibrium strategies y_B^*, y_G^* . In particular, if the optimal credential levels differ, $y_B^* \neq y_G^*$, then if observed $y = y_B^*$, $\beta^*(1 | y) = 1$ and if one observes $y = y_G^*$, $\beta^*(1 | y) = 0$. Of course, in the case when the optimal credential levels coincide, $y_B^* = y_G^*$, then if observed credential $y = y_B^* = y_G^*$, $\beta^*(1 | y) = 1 - q$. Since the predictive power of PBE is weak in a sense that it does not restrain the out of equilibrium beliefs, in the rest of the paper we use the Cho-Kreps (1987) intuitive criterion, and focus on the least cost separating (lcs) equilibrium.

Next, we analyze the optimal behavior of programmers and demonstrate the implications of externalities. When individual programmers decide on the level of credential they acquire in order to signal their skills, they do this by anticipating the wage offer of the software company. Programmers thus choose the optimal credential levels y_G^* and y_B^* by maximizing their utilities U_G and U_B given the wage function (1)

The incentive compatibility constraints for "bad" and "good" programmers read as

$$M(1 + ky_B) - y_B \geq M(2 + ky_G) - y_G, \quad (2)$$

$$M(2 + ky_G) - \frac{y_G}{2} \geq M(1 + ky_B) - \frac{y_B}{2}. \quad (3)$$

In order to focus on the least cost separating equilibrium with standard properties we constrain our analysis to cover the case when the magnitude of externalities $k \in (-\infty < k \leq \frac{1}{2M}]$. That is, we consider the possibility of a substitute OS program, and a not too strong complement OS program¹³.

¹³For the analysis of the equilibria for large positive externalities, $k > \frac{1}{2M}$, consult Leppämäki and Mustonen (2004) where the authors show that the nature of least cost separating equilibria may be quite different compared to the "normal" case analyzed in the literature and here as well.

We focus on the least cost separating equilibrium, $y_B^* \neq y_G^*$. Given the software companies's beliefs β^* and wage function w^* , and since getting a credential is costly, it is optimal for the "bad" programmer not to get one

$$y_B^* = 0.$$

The "good" programmer chooses the lowest level of credential that allows him to separate from the "bad" programmer. The level of credential the "good" programmer will choose can be solved from the "bad" type's binding IC -constraint (2)

$$y_G^* = \frac{M}{1 - Mk}.$$

Consequently the wages are $w_B^* = M$, $w_G^* = M \left(2 + k \frac{M}{1 - Mk} \right)$. In the least cost separating equilibrium, the utilities of the programmers are

$$U_B^* = M,$$

$$U_G^* = 2M + \frac{(2kM - 1)M}{2(1 - Mk)}.$$

As we have now derived the least cost credential levels, we can state the following result:

Proposition 1 *The least-cost OS programming credential level that separates the "good" programmer from the "bad" programmer in the case of a substitute (a complement) OS program is lower (higher) than in the case when the OS program is independent, $y_G^{*s} < y_G^{*i} < y_G^{*c}$.*

Proof. It is enough to notice that in the case of substitutes ($k < 0$), $y_G^{*s} = \frac{M}{1 - Mk}$ is smaller than than $y_G^{*i} = M$ in the case of independent programs ($k = 0$), which in turn is smaller than $y_G^{*c} = \frac{M}{1 - Mk}$ in the case of complements ($k > 0$). ■

When the OS programming to attain a credential creates a substitute to the monopolist's commercial program, the high productivity programmer suffers from this. The competition in the consumer market lowers the

monopolist's profits and thus ultimately the wage the "good" programmer receives once he is hired by the firm. Thus the programmer optimally adjusts the credential level downwards. In the case of complementary goods i.e. with positive market externalities the programmer internalizes the positive externality effect via the increased wage bill, and thus optimally adjusts the credential level upwards.

It is also useful to notice that the level of credential the "good" type has to choose in order to signal credibly his type increases in the magnitude of market externality, and this of course is also reflected in the utility as well. In short, this simply means that given the choice over the type of an OS project, the good "type" always prefers to go for OS projects that create complements to commercial software. Interestingly this may not always be the action preferred by the social planner. We come back to this point later on in section 5. Finally it should not come as a surprise that if we assume away market externalities i.e. set $k = 0$ and normalize market size effect by setting $M = 1$, the results we derive above coincide with those of Spence (1973, 1974).

4 Screening with OS Programming

From previous section we have learned that in the PBE, commercial software companies beliefs about the productivity of programmers are consistent, and thus their behavior of hiring programmers with high programming credential and paying them high wages is fully rational. In this section, we consider the incentives of commercial software companies to support OS projects financially.

In the following, we make a simplifying assumption that there exists an outside employment option with wage w_o , and this outside option is accessible for both the low and high productivity programmers. To rule out pooling equilibria, we want to examine now whether OS projects can function as screening devices for commercial software companies. To do so we assume that the level of outside wage is such that the software company is unprofitable if it hires a programmer blindly. That is

$$R - w_o = M(1 + q) - w_o < 0.$$

Simply to exist in the first place, the software company has to screen programmers and then by hiring the high productivity programmer it makes a positive market revenue. As before, since the programmer holds the bargaining power in the labor market, he receives all the revenue as wages. In particular, we want to demonstrate that under these conditions it is in the commercial software company's interest to create and support open source programming projects and use them as screening devices.

For simplicity, we assume that the support for open source projects is materialized in the form of a lump-sum payment F . Then the profit of the commercial software company reads as $\pi = M(2 + ky_G) - F - w = 0$, where y_G is determined by the bad programmer's incentive compatibility constraint with respect to the outside wage w_o ,

$$M(2 + ky_G) - F - y_G \geq w_o. \quad (4)$$

The least-cost separating level of programming credential is

$$y_G^* = \frac{2M - w_o - F}{1 - Mk} \quad (5)$$

From (5) we can see that the software company can provide financial support for the OS project up to the difference between the market revenue from its program created by a good programmer and the outside wage, $F < 2M - w_o$. Quite naturally the least cost separating credential is then decreasing in F . We can summarize the discussion as follows:

Lemma 2 *The commercial software companies do have incentives to support open source software projects in order to use them as screening devices for their new labor force.*

Perhaps one can even interpret this type of practices as outsourcing of personnel management and recruiting activities. As such it is rather clever managerial practice of motivating potential employees to signal their capability without committing to hire anyone at that stage. It is useful to

stress that in practise the support for OS projects may take various of forms such as advice, sharing of information, providing technical support, and so on. One should notice, that we are abstracting away from possible free-riding problems among different software companies that might arise within such an arrangement. Clearly, when two or more commercial software companies are present and supporting the same OS project there is a possibility that the competitor succeeds in hiring top programmers even though all companies were supporting the OS project at the outset. However, this should not change our reasoning qualitatively since it would only reduce the expected profits of software companies.

5 The Choice of An OS Project: Programmer's vs. Social Incentives

In section 3 we concluded that if the programmer can decide how to allocate his attention in terms of attaining a credential it is clear that he will devote attention towards the signaling activity that will produce as by-product complementary software, since his utility in the least cost equilibrium is increasing in k ; $\frac{\partial U_G^*}{\partial k} > 0$. In this section we derive conditions under which this may be in conflict with the society's interests.

In order to examine private and social incentives we need to first define the expression for the social welfare. The signaling activity as such is naturally costly for the programmer, but the resulting freely available software is valued by consumers, and clearly this is valuable for the society as well. To get a slightly more general view of the comparisons of private and social incentives we now characterize programmers with productivities $\theta_G > \theta_B > 0$ instead of 2 and 1. This is simply due to the fact that now the productivity difference does matter. When the software company employs the "good" programmer, the social welfare, measured as the net of market surplus, w_C or w_S ¹⁴, and the cost of signaling, in the case of a complement good, $k > 0$,

¹⁴The market surplus in the case of a complement consists of the firm's revenue, $M(\theta_G + ky_G)$, and the consumer surplus accruing to the buyers of the firm's program, $\frac{1}{2}M(\theta_G + ky_G)$. Thus $w_C = \frac{3}{2}M(\theta_G + ky_G)$. For a substitute, the surplus consists of

is

$$W_C = w_C - \frac{y_G}{\theta_G} = \frac{3}{2}M(\theta_G + ky_G) - \frac{y_G}{\theta_G}. \quad (6)$$

And in the case of a substitute, $k < 0$, the social welfare is equal to

$$W_S = w_S - \frac{y_G}{\theta_G} = \frac{3}{2}M\left(\theta_G - \frac{1}{3}ky_G\right) - \frac{y_G}{\theta_G}. \quad (7)$$

It's useful to recall from earlier analysis that in the case of substitute programs, profits and thus wages are decreasing in k but the social welfare as captured by (7) may increase in k . This is due to the fact that those consumers that find the commercial program too expensive acquire instead the freely available OS software.

As in previous section we assume that there exists an outside option equal to w_o , and set $F = 0$ in which case equation (4) yields

$$M(\theta_G + ky_G) - \frac{y_G}{\theta_B} \geq w_o,$$

which implies that the least cost programming credential that the "good" type chooses to separate is

$$y_G^* = \frac{\theta_B(M\theta_G - w_o)}{1 - Mk\theta_B}. \quad (8)$$

The social welfare in the case of a complementary OS program is obtained simply by inserting the expression (8) into (6):

$$W_C = \frac{3}{2}M\left(\theta_G + k\frac{\theta_B(M\theta_G - w_o)}{1 - Mk\theta_B}\right) - \frac{\theta_B(M\theta_G - w_o)}{\theta_G(1 - Mk\theta_B)}.$$

When the free OS program is a substitute, the social welfare as expressed in (7) yields after substitution

$$W_S = \frac{3}{2}M\left(\theta_G - \frac{1}{3}k\frac{\theta_B(M\theta_G - w_o)}{1 - Mk\theta_B}\right) - \frac{\theta_B(M\theta_G - w_o)}{\theta_G(1 - Mk\theta_B)}.$$

the firm's revenue and the consumer surplus to the buyers of the firm's program, in total $\frac{3}{2}M\theta_G$, and the surplus to the users of the free substitute OS program, $-\frac{1}{2}Mky_G$. Summed up, $w_S = \frac{3}{2}M(\theta_G - \frac{1}{3}ky_G)$.

Given these expressions, we fix the magnitude of the market externality effect equal to m in both cases. That is, $k = -m$ with a substitute and with a complement, $k = m$. This is done simply to facilitate the comparison, and we can thus examine under which conditions the society would prefer a substitute OS project compared to a complementary OS project that we know from above is always preferred by the individual programmer. That is, we consider now the possibility that society as such is able to choose the nature of an open source project. The straightforward comparison of above expressions gives us the following result on the *possible conflict of interest between the private and social incentives* over the choice of an OS project:

Proposition 3 *When the size of the market (given the the magnitude of externality) is small i.e. when $M < \frac{2\theta_B - \theta_G}{2\theta_G\theta_B m}$ programmers would prefer a complement, but the society would go for a substitute, and (ii) when the size of the market is large i.e. when $M > \frac{2\theta_B - \theta_G}{2\theta_G\theta_B m}$ the private and social incentives coincide, and both would prefer the OS project that produces a complementary software.*

Proof. To develop the above result we simply need to derive the inequality $W_C(k = m) < W_S(k = -m)$, since we know from earlier that programmers always prefer complementary OS projects. Developing the above inequality yields the following

$$\frac{3}{2}M\theta_G + \frac{3Mm}{2(1 - Mm\theta_B)} - \frac{1}{\theta_G(1 - Mm\theta_B)} <$$

$$\frac{3}{2}M\theta_G + \frac{Mm}{2(1 + Mm\theta_B)} - \frac{1}{\theta_G(1 + Mm\theta_B)},$$

which simplifies into

$$M < \frac{2\theta_B - \theta_G}{2\theta_G\theta_B m}.$$

■

In short, the above proposition shows that the possibility of potential conflict of interest over the choice of an OS project will depend most importantly on the size of the consumer market for software. The crucial point for understanding this result is related to the different market outcomes under

substitute and complementary programs. In particular, with a substitute OS program some consumers will buy the commercial programs and the rest will get the free OS software whereas with a complement only half of the consumers use the commercial program and the complementing free OS program. Since an individual programmer (and the commercial software company) does not internalize the beneficial effects that are coming via the consumer's surplus it is clear that when the consumer market for software are small there is a conflict of interest, and the society would prefer a substitute OS program to a complement. This conflict of interest, however, vanishes when the size of the consumer market increases.

When interpreting the above proposition from other direction it is not that easy to provide clear intuition for the result. However, we see that when consumers' valuation for the commercial program is low, the welfare associated with OS project resulting in a substitute program is higher. With our specifications, larger welfare with a substitute OS program is possible if the productivity difference between the programmers is small; the "good" programmer is less than twice as productive as the "bad" one.

Since we concluded that under some circumstances there is a potential conflict between the private and social incentives over the choice of an OS project, the natural follow-up question is whether the social optimum can be achieved by some sort of public policy intervention. In this respect, the answer that our analysis provides is rather straightforward, and points out towards direct subsidies for the OS projects that will lead to substitute programs when the consumer market for software is small.¹⁵

6 Conclusion

This paper introduced an idea of OS programming as a signaling and screening device. In particular, we proposed a skill signaling model that extended the celebrated Spence's model in to the situation where the signaling activity itself may have externality effects that are coming via the consumer

¹⁵See Schmidt and Schnitzer (2003) who discuss extensively the role of public subsidies for OS activities.

market for software. The size and magnitude of externality effect depends on whether the free OS software is either a substitute or a complement to the commercially produced one.

It was shown that "good" programmers have incentives to participate in OS projects in order to signal their programming ability to commercial software companies. Equally well, it was shown that it's fully rational for the software companies to interpret the high attained programming credential in OS projects of being a credible signal of programmers' innate ability and thus pay them higher wages. In addition, we demonstrated that the widely documented software companies' financial and other support for OS projects can be understood from screening perspective. By supporting OS projects the management of a software company is able to motivate potential employees to demonstrate their programming capability without committing to hire anyone at that stage.

Our focus on OSS is motivated by the 'virtuality' of software and the resulting possibly strong market responses to the availability of free OS software. Of course, it is important to realize that our analysis is partial in a sense that we focus on one market niche only. In reality, it may very well be that even if the OS software is independent in this particular market niche we examine, it may well have external effects on some other market niche. This is a natural question to be examined in future studies.

References

- [1] Bessen J., 2001: Open Source Software: Free provision of complex public goods, mimeo, www.researchoninnovation.org/opensrc.pdf, (Accessed April 2002), Research on innovation.

- [2] Bitzer J. and P. Schröder, 2002: Bug-Fixing and Code-Writing: The Private Provision of Open Source Software, Discussion Papers DIW Berlin, www.diw.de/deutsch/produkte/publikationen/diskussionspapiere/docs/papers/dp296.pdf (Accessed July 2004).

- [3] Browne C., 1999: The economics of free software, cb-browne.com/info/freecon.html (accessed June 2004).
- [4] Collab.net 2001: Open source networks, www.collab.net (accessed June 2004).
- [5] Dasgupta P. and P. David , 1987: Information disclosure and the economics of science and technology, CEPR Discussion Paper 73, also in: Feiwel D ed., Arrow and the ascent of modern economic theory (New York University Press, New York).
- [6] Dasgupta P. and P. David, 1994: Toward a new economics in science, *Research Policy* 23(5), 487-521.
- [7] Dewatripont M., I. Jewitt and J. Tirole, 1999: The economics of career concerns I&II, *Review of Economic Studies* 66(1), 183-217.
- [8] Fershtman C. and N. Gandal, 2004: The determinants of output per contributor in open source projects: An empirical examination, *CEPR, Discussion Paper* No. 4329.
- [9] GNU Project, 2000a: Categories of free and non-free software, www.gnu.ai.mit.edu/philosophy/categories.html (accessed June 2004).
- [10] GNU Project, 2000b: What is copyleft? www.gnu.ai.mit.edu/copyleft/copyleft.html (accessed June 2004).
- [11] Hann I., Roberts J., Slaughter S. and Fielding R., 2002: Delayed Returns to Open Source Participation: An Empirical Analysis of the Apache HTTP Server Project, Graduate School of Industrial Organization, Carnegie Mellon University, mimeo.
- [12] IBM News, 2001: www.ibm.com/news (Accessed December 2001).
- [13] IBM 2004: Linux executive report from IBM, <http://www-1.ibm.com/linux/whitepapers/index.shtml> (Accessed July 2004).
- [14] Johnson J., 2002: Open Source Software: Private provision of a public good, *Journal of Economics and Management Strategy*, 11(4), 637-662.

- [15] Koboldt C., 1995: Intellectual property and optimal copyright protection, *Journal of Cultural Economics* 19(2), 131-155.
- [16] Kogut B. and A. Metiu 2001: Open source software development and distributed innovation, *Oxford Review of Economic Policy* 17(2), 248-264.
- [17] Leppämäki, M. and M. Mustonen 2004: Signaling with Externality, *Helsinki Center of Economic Research (HECER), Discussion Paper*.
- [18] Lerner J. and J. Tirole 2002: Some Simple Economics of Open Software, *Journal of Industrial Economics* 50(2), 197-234.
- [19] Linux-PAM, 2003: <http://www.kernel.org/pub/linux/libs/pam/HallOfFame.html> (Accessed January 2004).
- [20] Mustonen M., 2003: Copyleft - the economics of Linux and other open source software. *Information Economics and Policy*, 15(1), 99-122.
- [21] Mustonen M. 2004: When do firms support the development of substitute copyleft programs? *Journal of Economics and Management Strategy*, forthcoming.
- [22] Netcraft, 2001: The netcraft web server survey, Netcraft, Bath, England.
- [23] Raymond E., 1998: The cathedral and the bazaar, www.firstmonday.dk/issues/issue3_3/raymond/index.html (Accessed April 2002).
- [24] Schiff A., 2002: The Economics of Open Source Software: A survey of the early literature, *The Review of Network Economics* 1(1), 66-74.
- [25] Schmidt, K. and M. Schmitzer 2003: Public Subsidies for Open Source? Some Economics Policy Issues of The Software Market, *Harvard Journal of Law and & Technology* 16. 473-505.
- [26] Spence, M. 1973: Job Market Signaling, *Quarterly Journal of Economics* 87. 355-74.

- [27] Spence, M. 1974: *Market Signaling*, Harvard University Press, Cambridge, USA.
- [28] Spence, M. 2002: Signaling in Retrospect and The Informational Structure of Markets. *American Economic Review* 92. 434-459.
- [29] Stern S. 1999: Do scientists pay to be scientists? *NBER Working Paper* 7410.