Economic Analysis of Open Source Software

By YU WANG

A dissertation submitted to

The Faculty of Social Sciences

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Economics and Management

University of Aarhus

Denmark

Table of Contents

Preface

This thesis was written during my studies as a PhD student at the School of Economics and Management, University of Aarhus in the period from September 2003 to September 2008. Some works were done during my visit to Marshall School of Business, University of Southern California.

First of all, I would like to thank the school for providing a good and stimulating research environment. Additionally, I am grateful to the school's generous financial support, which has allowed me to attend various courses and seminars, workshops, conferences and studying abroad.

Second, many thanks to my main thesis advisor, Anthony Dukes, for his continuing patience, kindness and encouragement. I greatly appreciate him always taking the time to help me with my research work during my PhD studies. I would also like to thank my secondary advisor, Per Balzter Overgaard, for his support and inspiration.

From February 2008 until August 2008, I had the great pleasure of spending time as a visiting scholar at Marshall School of Business, University of Southern California. I am very grateful to the school for its hospitality and inspiring environment.

I would also like to thank the faculty members and the staff at the School of Economics and Management, University of Aarhus, who have made research environment so pleasant to work in.

Finally, I would like to thank my family and friends both in Aarhus and China for their love, support and understanding throughout the years.

Yu Wang
Aarhus, October 2008

Summary

This thesis consists of three self-contained chapters that conduct theoretic analysis in the area of economics of open source software. The aim is to provide better understanding on the evolution of open source software and the interaction between open source software and business involvement. Although self-contained, these three chapters are interrelated. The first chapter explores the mechanism of the choices of open source licenses under the commercial involvement. The second chapter analyzes strategic reasons for profit-pursuing firms to adopt open source method. The last chapter studies the optimal licensing scheme for software companies when both proprietary and open source are available.

The open source software development seems to present a great challenge to the traditional economic theories with the facts that OSS development is heavily based on the contributions from many volunteer programmers and highly qualified software is produced within the loosely organized community and available at no cost, etc[Lerner & Tirole (2002)]. The most interesting thing is that this new developing method has made great success and become one main alternative of the present commercial software development method. For example, the OSS operating system Linux has obtained 38% market share in server operating system [Bitzer (2004)] and the OSS web server software Apache is even dominant with 50.76% market share in 2007. The OSS web browser Firefox also accounts for 16.8% market share in 2007. At the same time, more and more for-profit firms also come to embrace the open source software because of their popularity and success, including some software giants like IBM, SUN and HP.

Many research works have been done to study economics of open source software. Von Krogh & Von Hippel (2006) make a review of literature on open source in recent years. They categorize the research into three areas: motivations for contribution, governance, organization and innovation process, and competitive dynamics. The three papers in my thesis contribute to the third area. Specifically, my research work focuses on impact of open source software on software industry, influence of firms' involvement on open source software's evolution and melding of open source software and business.

The main focus of Chapter 1 is on the impact of the commercial involvement on the open source licensing scheme. When more and more commercial companies

are engaged to open source software development, the licensor of open source project must consider the how to choose an appropriate license to make the OSS project best supported not only by open source communities but also firms who want to build business upon open source software. When the open source community has preference over the copyleft licenses, the participation of for-profit firms will not change the dominance of copyleft licenses in the market. Instead, the involvement of more firms will reinforce their popularity.

Chapter 2 tries to explore strategic reasons for profit-pursuing firms to adopt open source method, that is, give away the source codes of their software products. It also tries to study how open source software impacts on software industry. When the firms compete for a group of heterogeneous consumers in the market, who need to buy software and enhancement sequentially in two stages, open source software may be invited by the incumbent to dominate the market if the usage of software has network externality and the value of enhancement is dependent on the adopt base of the software. Another surprising finding is that presence of open source software may relax the competition of software industry, contrary to the common belief that open source will always increase competition.

Chapter 3 presents a theoretical model to study the choice of optimal licensing scheme of a software company. Software companies have three alternative licensing scheme to develop and distribute the software product: pure open source, pure proprietary or mixed. The choice of different licensing scheme depends on the benefits of free contribution from the open source community and the profit from product differentiation by licensing. The analysis shows that pure open source licensing scheme is always preferred as long as the development cost is high. If the development cost is moderate, pure proprietary licensing is optimal when the users are support-independent. When the users are support-oriented, the choice between pure open source and mixed strategy depends on whether the users are sensitive to the reciprocal terms in the open source license. At last, this model also provides another explanation of licensing diversification of open source licensing.

References

Bitzer, J. (2004), 'Commercial versus open source software: The role of product heterogeneity in competition', *Economic Systems* **28(4)**, 369–381.

Bitzer, J. & Schroder, P. J. H. (2006), *The Economics of Open Source Software Development*, Elsevier.

Lerner, J. & Tirole, J. (2002), 'Some simple economics of open source', *Journal of Industrial Economics* **50**, 197–234.

Von Krogh, G. & Von Hippel, E. (2006), 'The promise of research on open source software', *Management Science* **52(7)**, 975–983.

CHAPTER 1

Choice of Open Source Software Licenses under Commercial Involvement

Choice of Open Source Software Licenses under Commercial Involvement

Yu Wang*
University of Aarhus

May 10, 2009

Abstract

This paper sets up a theoretical model to study how the commercial involvement affects the choice of open source licenses. When the open source community has preference over the copyleft licenses, the participation of for-profit firms will not change the dominance of copyleft licenses in the market. Instead, the involvement of more firms will reinforce their popularity. At the same time, I also find that the participation of commercial companies will lead to the free riding problem, which can not be prevented by the licensor. At last, the commercial involvement will make the licensor more dependent on the open source community.

JEL classifications: C73, D45, L17, L86

Keywords: copyleft, reciprocity terms, hijacking benefit, complementary benefit, *qui pro quo* effect

*Address correspondence: Yu Wang, Department of Management, Building 1322, University of Aarhus, DK-8000 Aarhus C, Denmark. E-mail: ywang@econ.au.dk

1 Introduction

With the popularity of the open source software (OSS) in the last decade, the open source has become a main alternative of the dominant commercial software development method. The open source movement also attracts many scholars to make effort to study this interesting phenomenon. Like the traditional commercial software, open source software is also distributed under the specific licensing terms. However, the motivation of the licensor's choice of licensing scheme is completely different between them. Licensing of commercial software can be regarded as an issue of managing the intellectual property right to get maximized profits. A number of literatures have made contributions into this field, such as Gallini (1984), Katz & Shapiro (1986) and Rockett (1990). The common key point of these papers is that using the intellectual property right as a competition tool to maximize the benefits of the right-holder. Since open source software is developed by many voluntary programmers, who freely contribute codes to the project, and also freely available to everyone, maximizing profits seems to be irrelevant to the licensor's choice of license terms. It means that a new method or new determinants should be considered when studying the mechanism of licensing OSS projects. And the study of licensing of open source software will definitely enrich the research of economics of technology licensing and shed light on some phenomenon in open source movement.

Early studies about open source licenses mainly focused on the description and classification of their features [Lee (1999) and Valimaki (2005)], but little has been done to study why the licensor chooses one particular open source license instead of another. **?** made the initial examination of this question. They presents an empirical analysis to explore the relationships between license choice and project characteristics. They argue that the licensor's choice will be affected by the preference of contributors besides his own, and both of them are various. However, they only consider the interaction between the licensor and the community, yet ignore the for-profit firms' involvement and its impact on the choice of open source licenses. Actually, more and more firms are engaged into open source software in recent years and have set up quite successful business models based on open source software. They gradually play more important roles in the open source movement. According to the sample in **?**, 82% of the open source projects are under the restrictive licenses. One question is raised that whether the commercial involvement will make the permissive licenses more popular or the restrictive licenses still be favored.

In this paper, I set up a model to explore how the licensor chooses the different reciprocity terms in the open source licensing to maximize the contributions from both open source community and for-profit firms. I found that when the community has preference for high reciprocity degree, the commercial involvement will not change the dominance of the restrictive licenses. However, the licensor is more likely to choose less restrictive license when the firm does earn positive but small profits from supporting the open source project. Another interesting result is that the restrictive license is always preferred when more than one firm are engaged. Section 2 reviews the relevant issues about open source software, such as motivations of contribution and open source licenses. A model is set up in Section 3 and Section 4 is the conclusions and further discussions.

2 Licenses of Open Source Software

The general definition of open source software is that a program is open source software if its source code is freely available[1]. Many people confuse open source software with the public domain software because both of them are publicly available. Unlike the public domain software, which is not copyrighted, open source software is always licensed and under the protection of copyright law [Bobko (2000)]. The open source license is an important characteristic of open source software and its birth can be regarded as a legal innovation [Bonaccorsi & Rossi (2003*b*)].

According to the copyright law, the copyright holder, or the author of code in the case of software development, is entitled the monopolistic and exclusive right to reproduce, modify and distribute, which give the copyright holder power to forbid any unallowed use. The proprietary software author use this privilege to obtain profits by selling permission to use the software. On the contrary, open source advocates want to encourage the free use of the software [Horne (2001)] So they also use the copyright law as weapon to protect the free distribution of

[1] To see more specific definition from Open Source Initiative (OSI), please visit http://www.opensource.org

open source software besides releasing source code.

Bessen (2004) argues that open source software is a privately produced public good. When contributors voluntarily contribute some code to the open source project, the software itself is a common code pool. They want to keep this common pool continue to grow instead of shrinking. Compared with the public domain software, contributors of open source software are the copyright holders and they do retain some privilege even they relax the control of the code they writing and give away most right entitled by the copyright law.

Almost all the open source licenses freely give away the right of use, modification and distribution. However, they are quite different in the terms of distribution of derivative works. Some of them impose reciprocity terms, that is, all the derivative works must be distributed under the same license term, which means it is impossible for one programmer to improve the open source code and privatize it [Fink (2003)]. These licenses include GPL and MPL. The first one is widely used in Linux system and the second one is the license of Firefox. Open source licenses with such reciprocity terms are also known as copyleft licenses. At the same time, some open source licenses are lack of such reciprocity terms on derivative works, such as BSD. It imposes nearly no restriction on redistribution and use, so the derivative code under BSD can be redistribute under any license scheme, open source or proprietary. This kind of licenses are called non-copyleft licenses.

3 Motivations of Open Source Software Contributors

Why are there so many talented programmers contribute to the open source software for free? Open source software's popularity and success have attracted many economists to study this phenomenon. Feller & Fitzgerald (2002) classifies the contributors' motivations into three categories: economic motivations, social motivations and technological motivations.

First let's discuss the economic motivations. Lakhani & Von Hippel (2003) find in their survey of Apache project, more than 80% of the posted questions only take five minutes in average to be solved. The Internet has make it almost costless for the skilled programmers to access a wide audience and help them to solve programming problems. At same time, the contributors can gain a reputation among peers by helping to solve difficult problems [Dalle & David (2003)] and also signal his programming ability, which is helpful to increase future career benefits [Lerner & Tirole (2002) and Lerner & Tirole (2004)].

Some contributors take part in open source also because of the social motivations. Stallma (1984) argues that everyone should have the right to use and modify the software and the freedom of software should not be restricted. Dissatisfaction with the monopoly of proprietary software is an important motivation behind the early open source movement. The other social motivations includes fun to program [Torvalds & Diamond (2002)] and self-identification and sense of belonging to the community [Raymond (2000*b*) and Raymond (2000*a*)].

The technological motivations are quite simple. Programmers also consider their contribution as a learning process and improve their programming skills [Lakhani & Von Hippel (2003)]. Kuan (2001) proposes that the lack of existing software to perform a particular task is another incentive for the programmers to create an open source project. Generally, the technological motivations can be summarized as the software programmer tries to use the open source platform to solve the specific technological problems.

These motivations can partially explain the puzzle of free contribution of voluntary programmers to the open source software. All individuals with above motivations can be called the community contributors because they all behave similarly, working for the open source project without any direct monetary reward, and they all have strong sense of belonging to the community. On the other hand, there are some programmers, whose involvements are directly motivated by the monetary rewards. Hertel, Niedner & Herrmann (2003) find in their survey that about 20% of developers of the Linux kernel receive salary on a regular basis for their work contributed to the Linux projects. These people are obviously employees of other firms but their employers pay them to work for the open source projects, which are freely available. At the same time, some software companies directly provide financial support to the open source projects, such as IBM, Oracle and Sun. So now these firms can be regarded as the other important force on supporting the open source movement. Firms are also have various motivations to support open source projects, which includes obtaining indirect revenues from selling related products [Hecker (1999)], Lowering innovation cost [Hawkings (2004)] and others. However, their contributions are only for one aim in the end: obtaining economic profits, just like their other investment strategies. Bonaccorsi & Rossi (2003*a*) found in their survey that firms contribute to the open source projects mainly for economic and technological reasons.

4 Model

4.1 One Single Firm As Potential Contributor

Suppose an original developer plans to develop a new software programme. He releases the original code under the open source license. **?** argue that the licensor needs to consider various benefits when he selects the license. Here, I just make a simple assumption about the licensor.

Assumption 1: The licensor tries to maximize the effort contributed to the open source project from all contributors.

Even though the licensor may have quite various motivations to launch an open source project, it is reasonable and safe to presume that his utility should be positively related to the growth of the project or the common code pool. The software quality will be improved if more efforts are continuously contributed to the project, which will definitely bring the licensor, mostly also the leader of the project, more benefits. These benefits include stronger signal effect with large adoption, greater potential material benefits, or better solutions to some specific problems.

Most open source licenses are quite similar except the reciprocity term, so I assume the licensor only needs to choose an appropriate reciprocity degree $\gamma \in [0, 1]$. When $\gamma \to 1$, the license is more similar to BSD thus quite permissive. While $\gamma \to 0$, the license is restrictive on the derivative works, like GPL.

The development of the open source project goes through two stages: the early stage and the mature stage. In each stage, the project receives contributions from two parties: the open source community and the for-profit firm. They choose their contributions simultaneously in each stage, which can be observed by each party in the end of each stage.

As discussed earlier, the for-profit firm has quite simple and pure motivation to contribute to the open source project: obtaining profits. However, the community has quite heterogeneous motivations since it is composed by many voluntary programmers with different incentives. The difference of motivations do affect the preference for the reciprocity degree of license terms. According to the surveys [Bates, Di Bona, Lakhani & Wolf (2002) and Hars & Ou (2002)], the following three motivations are considered to be the underlying and majority reasons for individual programmers in the community to contribute to the open source projects: learning and improving skills, software freedom, sense of belonging to the open source community. Since higher reciprocity degree can provide better protection from hijacking[2] and hijacking will do harm to the interests of the contributors with above three motivations, the open source community will naturally prefer the license with high reciprocity terms. But the firm has the opposite preference, that is, he can earn more profits when the open source project is licensed without reciprocity terms because there is no restriction on how he can deal with the open source code and its derivative works. Now I make an assumption about the community.

Assumption 2: The open source community prefers the open source license with the reciprocity terms.

The community's utility in the first stage is

$$U_C^1 = a - \left[e_0^1 - \left(1 - \frac{\gamma}{2}\right)\right]^2$$

and the utility in the second stage is

$$U_C^2 = \begin{cases} a - (e_0^2 - \frac{1}{2})^2, \text{ if } \gamma \neq 0 \text{ and } e_1^1 = 0 \\ a - (e_0^1 - 1)^2, \text{ otherwise} \end{cases}$$

where $a > 0$ and $e_0^t \in [0, 1]$ is the contribution from the community in stage $t = 1, 2$ and similarly $e_1^t = \{0, 1\}$ is the contribution from the firm in stage t.

According to the Assumption 2, the community favors high reciprocity degree, which can provide better protection from hijacking. So their contribution in the early stage is always increased with γ. At the same time, they also need to observe the action of the firm. If the firm also contributes in the early stage, the community is also willing to increase their contribution in the second stage even the license is permissive. I call it *quid pro quo* effect.

[2] Hijacking is a term used by Lerner and Tirole (2002) to describe the situation where the commercial firm privatize the open source code.

The firm's utility in each stage is

$$\begin{aligned} U_F^t &= U_H^t + U_S^t \\ &= \left(\gamma e_0^t\right) + \left(\alpha_t e_1^t - e_1^t\right) \end{aligned}$$

where

$$\alpha_t = \alpha \sum_{i=0}^{1} \sum_{r=1}^{t} (e_i^r)$$

and $\alpha > 0$.

The firm's utility is composed by two part: hijacking benefit and complementary benefit, which are represented by U_H^t and U_S^t respectively. Since the open source project is a common code pool, the firm can copy the existing code into their own proprietary software product to earn profits by redistribution. This privatization is also called hijacking. Hijacking benefit is greatly determined by the reciprocity degree of the license term, that is, it is decreased to 0 when γ changes from 1 to 0. The firm can always enjoy hijacking benefit as long as the license is not restrictive, in other words, $\gamma > 0$. On the other hand, the firm can also choose to support the open source project by making contribution to improve the code quality thus obtaining benefit from selling complementary service or accessories. I call it the complementary benefit.

Obviously the complementary benefits is determined by two factors: the potential business value and the code quality of the project. The potential business value is an important characteristic of the project, depending on what kind of functionalities it is programmed to realize. For example, operating system software is always believed to has high business value. At the same time, the evolution of business model related to the open source software also plays an significant role. Two decades ago, few people knew how to profit from software if it was freely available. With the deeper understanding of open source software and the innovation of business, many firms have made a great success from open source projects.

How to turn the potential business value into real profits? High business value never means great profits. Today's software market is quite competitive, code quality is the key to survive and succeed in the intensive competition. In the case of open source software, contribution from both parties has a dominant influence on the improvement of code quality. High code quality is the guarantee of large adoption base, which eventually contributes to the business profits because of the network effect of software use. So the firm normally has high risk if he chooses to invest the open source project in the early stage because the code quality still needs improvement.

Given the reciprocity degree γ chosen by the licensor, the firm has four possible strategies: full investment ($e_1^1 = e_1^2 = 1$), staying out ($e_1^1 = e_1^2 = 1$), early investment ($e_1^1 = 1$ and $e_1^2 = 0$) and late investment ($e_1^1 = 0$ and $e_1^2 = 1$).

Lemma 1 *Both early investment and late investment are dominated strategies.*

Proof. Suppose the firm adopts fully investment strategy, then his contributions in two stages are $e_1^1 = e_1^2 = 1$. Correspondingly the contributions from the community are $e_0^1 = 1 - \frac{\gamma}{2}$ and $e_0^2 = 1$ respectively in each stage. Then the total utility of firm is

$$\begin{aligned} U^F &= \left[\gamma(1-\gamma) + \alpha\left(2 - \frac{\gamma}{2}\right) - 1\right] + \\ &\quad \left[\gamma + \alpha\left(4 - \frac{\gamma}{2}\right) - 1\right] \\ &= 2\gamma - \frac{\gamma^2}{2} + \alpha(6-\gamma) - 2 \end{aligned}$$

Similarly the firm's utility from other three strategies are

$$\begin{cases} U^S = \frac{3}{2}\gamma - \frac{\gamma^2}{2}, \text{ staying out} \\ U^E = 2\gamma - \frac{\gamma^2}{2} + \alpha\left(2 - \frac{\gamma}{2}\right) - 1, \text{ early investment} \\ U^L = \frac{3}{2}\gamma - \frac{\gamma^2}{2} + \alpha\left(\frac{5}{2} - \frac{\gamma}{2}\right) - 1, \text{ late investment} \end{cases}$$

If early investment is the dominant strategy, it must satisfy

$$U^E \geq \max\left\{U^F, U^S, U^L\right\}$$

that is

$$\begin{cases} \gamma(1-\alpha) \geq 2 - 4\alpha \\ \alpha\gamma \geq 8\alpha - 2 \\ \gamma \geq \alpha \end{cases}$$

There is no such a pair of (α, γ) can satisfy these three inequations simultaneously when $\alpha \in (0, +\infty)$ and $\gamma \in [0, 1]$. So early investment is a dominated strategy.

Similarly, it is also easy to prove that late investment is also a dominated strategy. ■

So now the firm only have two optimal choices: either investing the open source project in both stages or not investing at all. By comparing the utilities under these two strategies and considering that the licensor will choose the appropriate reciprocity degree to maximize the contributions, I make the following proposition.

Proposition 1 *The licensor will choose* $\gamma = 0$ *when* $\alpha < \frac{3}{10}$*;* $\gamma = \frac{4-12\alpha}{1-2\alpha}$ *when* $\alpha \in \left[\frac{3}{10}, \frac{1}{3}\right]$ *and* $\gamma = 0$ *when* $\alpha > \frac{1}{3}$*. The firm will adopt the strategy of staying out if* $\alpha < \frac{3}{10}$ *but full investment if* $\alpha \geq \frac{3}{10}$*.*

Proof. From Lemma 1 we know that the firm will consider two strategies: full investment and staying out. So we only compare U^F with U^S.

If the firm's optimal strategy is full investment then the following conditions must be satisfied

$$\begin{cases} U^F \geq U^S \\ 0 \leq \gamma \leq 1 \end{cases}$$

At the same time, since the licensor will choose γ to maximize the total contribution and the firm's contribution in each stage are $e_1^1 = e_1^2 = 1$ in this case, it is easy to conclude that the licensor will always choose the minimum γ satisfying the above conditions. The solutions are

$$\begin{cases} \gamma = \frac{4-12\alpha}{1-2\alpha} \text{ when } \alpha \in \left[\frac{3}{10}, \frac{1}{3}\right] \\ \gamma = 0 \text{ when } \alpha > \frac{1}{3} \end{cases}$$

Similarly if the firm's optimal strategy is staying out, that is, $U^F \leq U^S$ and $0 \leq \gamma \leq 1$, then the licensor will always choose γ to maximize the contributions from the community because of no investment from the firm at all. The solution is

$$\gamma = 0 \text{ when } \alpha < \frac{1}{3}$$

■

Some interesting results from Proposition 1 are worth further discussion. The potential value of the open source project do have influence on the licensor's choice of license. When the project has high potential business value, the licensor know that the firm is always willing to contribute, so his priority will focus on how to convince the community to make the maximal contribution, which leads to higher reciprocity degree of license terms. On the other hand, the low business value has the similar effect but the difference is that the firm will not contribute at all and the community becomes the only source of contribution. The most interesting case is when $\alpha \in \left[\frac{3}{10}, \frac{1}{3}\right]$, specifically the potential business value is acceptable but not high enough to invest the project in both stages. At the first stage, the firm's complementary benefit is negative because the low business value. However, the complementary benefit at the second stage is positive if the firm keeps on investing in both stages. Now the licensor will try to relax the reciprocity term to increase the firm's hijacking benefit to motivate the firm to invest at the first stage. The reciprocity degree is decreased when the potential business value is lower but still acceptable. Now it is obvious that the involvement of business dose influence the licensing scheme. When the licensor only faces the community, the restrictive license will always be preferred because of the community's favor. But when another firm is engaged, the licensor may be inclined to choose the permissive license to attract the firm to contribute.

In the reality, open source projects facing only one single firm's involvement generally refer to those projects have close relation with some specific firms. They are either directly launched by the firm or the functionalities of the projects are complementary to the existing commercial software. Bonaccorsi & Rossi (2005) conducted a survey on Italian firms that do business with open source software. They found that 18.9% of firms exclusively choose permissive license scheme, but 81.1% of them work with open source software licensed under the restrictive licenses. These results are consistent with the prediction of my model. The license with high reciprocity terms will be dominant in the market, but under some situation some projects are more likely to choose permissive license to attract the firm's support.

4.2 More Than One Firms As Potential Contributors

Now Let's consider the case where there are more than one firms as the potential contributors. The increase of number of investors will inevitably increase the possibility of problems of free riding. One firm contributes to the open source project at his own cost but the improvement of the project's code quality has spillover effect. Other firms will enjoy the increased complementary benefit and hijacking benefit (the latter depends on the license term) without paying any costs. How does this free riding problem caused by the increasing number of firms influence the licensor's choice of license scheme? I will extend the original model above to discuss this issue.

Suppose the game's setting is the same as before but another firm is added to the original game and two firms has symmetric utility functions and action space. The firm j's utility in each stage is

$$\begin{aligned} U_{jF}^{t} &= U_{jH}^{t} + U_{jS}^{t} \\ &= \left[\gamma\left(e_0^t + e_j^t\right)\right] + \left(\alpha_t e_j^t - e_j^t\right) \end{aligned}$$

where

$$\alpha_t = \alpha \sum_{i=0}^{2} \sum_{r=1}^{t} (e_i^r)$$

and $\alpha > 0$, $e_j^t = \left\{0, \frac{1}{2}\right\}$ where $j = 1, 2$ and $t = 1, 2$.

The community's utility function is also similar. Its utility in the first stage is

$$U_C^1 = a - \left[e_0^1 - \left(1 - \frac{\gamma}{2}\right)\right]^2$$

and the utility in the second stage is

$$U_C^2 = \begin{cases} a - (e_0^2 - \frac{1}{2})^2, \text{ if } \gamma \neq 0 \text{ and } e_1^1 + e_2^1 = 0 \\ a - (e_0^1 - 1)^2, \text{ otherwise} \end{cases}$$

where $a > 0$ and $e_0^t \in [0, 1]$ where $t = 1, 2$.

To solve this game for equilibria, I use the method of backward induction. First let's look at the second stage. Two firms' payoffs in the second stage are described as the following:

		Firm 2	
		Contribute	Not Contribute
Firm 1	Contribute	(A, A)	(B, C)
	Not contribute	(C, B)	(D, D)

where $\begin{cases} A = \gamma\left(e_0^2 + \frac{1}{2}\right) + \alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + 1\right)\frac{1}{2} - \frac{1}{2} \\ B = \gamma e_0^2 + \alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + \frac{1}{2}\right)\frac{1}{2} - \frac{1}{2} \\ C = \gamma\left(e_0^2 + \frac{1}{2}\right) \\ D = \gamma e_0^2 \end{cases}$

There are three possible situations in the second stage which are discussed in the following:

Case 1: $\alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + \frac{1}{2}\right) \geq 1$

Then $A \geq C$ and $B \geq D$. In the second stage, there is one pure Nash Equilibrium: both firms contribute ($e_1^2 = e_2^2 = \frac{1}{2}$).

Case 2: $\begin{cases} \alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + \frac{1}{2}\right) < 1 \\ \alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + 1\right) \geq 1 \end{cases}$

Then $A \geq C$ and $B \leq D$. In the second stage, there are two pure Nash Equilibria: both firms contribute ($e_1^2 = e_2^2 = \frac{1}{2}$) or neither of the two firms contributes ($e_1^2 = e_2^2 = 0$).

Case 3: $\alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + 1\right) < 1$

Then $A \leq C$ and $B \leq D$. In the second stage, there is one pure Nash Equilibrium: neither of the two firms contributes ($e_1^2 = e_2^2 = 0$).

After the analysis of the second stage, let's turn to the first stage. Two firms' payoffs in the first stage are described as the following:

		Firm 2	
		Contribute	Not Contribute
Firm 1	Contribute	(A', A')	(B', C')
	Not contribute	(C', B')	(D', D')

where $\begin{cases} A' = \gamma\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) + \alpha\left(1 - \frac{\gamma}{2} + 1\right)\frac{1}{2} - \frac{1}{2} \\ B' = \gamma\left(1 - \frac{\gamma}{2}\right) + \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right)\frac{1}{2} - \frac{1}{2} \\ C' = \gamma\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) \\ D' = \gamma\left(1 - \frac{\gamma}{2}\right) \end{cases}$

Suppose now that Case 1 holds in the second stage, that is,

$$\alpha\left(1 - \frac{\gamma}{2} + e_1^1 + e_2^1 + e_0^2 + \frac{1}{2}\right) \geq 1$$

Since $e_1^1 + e_2^1 + e_0^2 \geq \frac{1}{2}$, it is easy to reach that

$$\alpha\left(1 - \frac{\gamma}{2} + 1\right) \geq 1 \tag{1}$$

When inequality (1) holds, $A' \geq C'$. So now there are two possible situations in the first stage:

1) If $\alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) \geq 1$, then there is a pure Nash Equilibrium of the game, that is, both firms contribute in both stages ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = \frac{1}{2}$). By solving

$$\begin{cases} \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2} + \frac{1}{2} + 1 + \frac{1}{2}\right) \geq 1 \\ \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) \geq 1 \end{cases}$$

we can get that

$$\gamma \leq \frac{3\alpha - 2}{\alpha} \tag{2}$$

2) If

$$\begin{cases} \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) < 1 \\ \alpha\left(1 - \frac{\gamma}{2} + 1\right) \geq 1 \end{cases}$$

then there are two pure Nash Equilibria of the game: both firms contribute in both stages ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = \frac{1}{2}$) or neither of firms contributes in the first stage but both of them contribute in the second stage ($e_1^1 = e_2^1 = 0$, $e_1^2 = e_2^2 = \frac{1}{2}$). Solving

$$\begin{cases} \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) < 1 \\ \alpha\left(1 - \frac{\gamma}{2} + 1\right) \geq 1 \\ \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2} + \frac{1}{2} + 1 + \frac{1}{2}\right) \geq 1 \end{cases}$$

or

$$\begin{cases} \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2}\right) < 1 \\ \alpha\left(1 - \frac{\gamma}{2} + 1\right) \geq 1 \\ \alpha\left(1 - \frac{\gamma}{2} + \frac{1}{2} + \frac{1}{2}\right) \geq 1 \end{cases}$$

will lead to the same result, which is

$$\frac{3\alpha - 2}{\alpha} < \gamma \leq \frac{4\alpha - 2}{\alpha} \tag{3}$$

Suppose now Case 2 holds in the second stage. Because $\frac{1}{2} \leq e_1^1 + e_2^1 + e_0^2 \leq 2$, we can get that

$$\begin{cases} \alpha\left(1-\frac{\gamma}{2}+\frac{3}{2}\right)<1 \\ \alpha\left(1-\frac{\gamma}{2}+\frac{5}{2}\right)\geq 1 \end{cases} \tag{4}$$

When inequality (4) holds, $A' \leq C'$ and $B' \leq D'$. So there is only one possible situation.

1) If $\alpha\left(1-\frac{\gamma}{2}+1\right) \leq 1$, there are two pure Nash Equilibria of the game: neither of the firms contributes in the first stage but both of them contribute in the second stage ($e_1^1 = e_2^1 = 0$, $e_1^2 = e_2^2 = \frac{1}{2}$) or neither of them contributes in either stage ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = 0$). Solving

$$\begin{cases} \alpha\left(1-\frac{\gamma}{2}+1\right)<1 \\ \alpha\left(1-\frac{\gamma}{2}+\frac{3}{2}\right)\geq 1 \\ \alpha\left(1-\frac{\gamma}{2}+1\right)\leq 1 \end{cases}$$

to get that

$$\frac{4\alpha-2}{\alpha} < \gamma \leq \frac{5\alpha-2}{\alpha} \tag{5}$$

At last suppose Case 3 holds, combining $e_1^1 + e_2^1 + e_0^2 \leq 2$ with

$$\alpha\left(1-\frac{\gamma}{2}+e_1^1+e_2^1+e_0^2+1\right)<1$$

it is easy to reach that

$$\alpha\left(1-\frac{\gamma}{2}+3\right)<1 \tag{6}$$

When inequality (6) holds, $A' \leq C'$ and $B' \leq D'$. So there is only one possible situation.

1) If $\alpha\left(1-\frac{\gamma}{2}+1\right) \leq 1$, there is one pure Nash Equilibria of the game: neither of the firms contributes in either stage ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = 0$). Solving

$$\begin{cases} \alpha\left(1-\frac{\gamma}{2}+\frac{3}{2}\right)<1 \\ \alpha\left(1-\frac{\gamma}{2}+1\right)\leq 1 \end{cases}$$

leads to

$$\gamma > \frac{5\alpha-2}{\alpha} \tag{7}$$

Combining inequalities (2), (3), (5) and (7), the following proposition can be reached.

Proposition 2 *It is always optimal for the licensor to choose* $\gamma = 0$.

1) When $\alpha \geq \frac{2}{3}$*, there is only one pure Nash Equilibrium that both firms contribute in both stages* ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = \frac{1}{2}$)*;*

2) When $\frac{1}{2} \leq \alpha < \frac{2}{3}$*, there are two pure Nash Equilibria of the game: both firms contribute in both stages* ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = \frac{1}{2}$) *or neither of firms contributes in the first stage but both of them contribute in the second stage* ($e_1^1 = e_2^1 = 0$, $e_1^2 = e_2^2 = \frac{1}{2}$)*;*

3) When $\frac{2}{5} \leq \alpha < \frac{1}{2}$*, there are two pure Nash Equilibria of the game: neither of the firms contributes in the first stage but both of them contribute in the second stage* ($e_1^1 = e_2^1 = 0$, $e_1^2 = e_2^2 = \frac{1}{2}$) *or neither of them contributes in either stage* ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = 0$)*;*

4) When $\alpha < \frac{2}{5}$*, there is only one pure Nash Equilibrium of the game: neither of the firms contributes in either stage* ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = 0$)*.*

Proof. When $\alpha \geq \frac{2}{3}$, $0 \leq \gamma \leq 1 \leq \frac{3\alpha-2}{\alpha}$. So inequality (2) always holds, and there is only one pure Nash Equilibrium that both firms contribute in both stages ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = 1$). The licensor will choose $\gamma = 0$ to maximize the total contribution $\left(1 - \frac{\gamma}{2} + 3\right)$.

When $\frac{1}{2} \leq \alpha < \frac{2}{3}$, $\frac{3\alpha-2}{\alpha} < 0 \leq \gamma \leq \frac{4\alpha-2}{\alpha} < 1$. So inequality (3) always holds, and there are two pure Nash Equilibria of the game: both firms contribute in both stages ($e_1^1 = e_2^1 = e_1^2 = e_2^2 = \frac{1}{2}$) or neither of firms contributes in the first stage but both of them contribute in the second stage ($e_1^1 = e_2^1 = 0$, $e_1^2 = e_2^2 = \frac{1}{2}$). It is always optimal for the licensor to choose $\gamma = 0$ to maximize total contribution, which is equal to either $\left(1 - \frac{\gamma}{2} + 3\right)$ or $\left(1 - \frac{\gamma}{2} + \frac{3}{2}\right)$.

The rest of the proposition can be proved in the similar way. ■

By comparing Proposition 1 and Proposition 2, we can find some interesting results. First, the dominance of restrictive license is even strengthened when another firm joins in. It implies that the more firms are attracted to join open source movement because of its popularity, the more likely the restrictive licenses are adopted by the licensor. However, the dominance of licenses with high reciprocity degree never means the licensor is better off when more firms are involved in. Actually, it is more difficult for the licensor to motivate the firms to support the open source projects because the licensor's ability of manipulation by licensing has been weakened by the free riding problems, which appears with another firm's involvement. When the licensor only faces one firm, he can always motivate the firm to support the project in both stages as long as its potential business value is acceptable ($\alpha \geq \frac{3}{10}$). However, when two firms are engaged into the project, this goal can only be achieved when the potential business value is quite high ($\alpha \geq \frac{2}{3}$). Otherwise it is possible that both firm postpone their support to the second stage or even neither of them contributes to the project at all in both stages. Free riding problem make both firms less willing to invest in the early stage and prefer to wait for the project to be mature by contribution from the community. When $\frac{2}{5} \leq \alpha < \frac{2}{3}$, the licensor can use high reciprocity degree to eliminate the hijacking benefit to relieve the free riding effect, but he can never get rid of it. In other word, the licensor can not prevent the possibility of delayed support or even no support from the firms.

5 Conclusion

In the recent years, the popularity of open source software and the success of related business models have attracted many commercial companies set foot in the open source software. Firms come to play a more important role in the development of open source projects compared with the period where only hackers take part in. The licensor now must consider this factor when he chooses licenses for the open source project. Besides the preference of the community, he also needs to take the firms' benefits into account to motivate their support. The licensor must seek balance between provide enough protection to the community to get free contribution from them and license the project in a way that the firms can get profits from their investment to the project.

In this paper, I set up a theoretical model to study how the commercial involvement affects the choice of open source licenses. When the open source project has low business value, its development is still supported by the community and the firm will not involve in. So the licensing scheme of these projects are unchanged. The community's preference determines the choice of license, that is, copyleft licenses. When the project's business value is increased but below some level, the licensor is more likely to choose non-copyleft license to obtain the support from the firm. However, copyleft license will be preferred again when the project's potential business value continue to increase or more firms join in.

On the whole, when the community has preference over license with reciprocity terms, the participation of for-profit firms will not change the dominance of copyleft licenses, for example the GPL. And this dominance will be reinforced when the new business model are invented to fully exploit the potential business value of open source software and more firms join this business.

On the other hand, the participation of commercial companies do bring challenge to the licensor, especially when more than one firms take part in. The adjusting capability of the licensing is undermined because now the licensor needs to find the appropriate reciprocity degree to balance the benefit not only between the community and the firms but also between firms themselves. The choice of reciprocity degree can not totally solve the free riding problem caused by increasing number of firms. The licensor can not prevent the firms from postponing or even canceling their investment to the project by purely using license. The interesting thing is that the licensor is more dependent on the support from the community when more firms are potential investors. It is unnecessary to worry that the involvement of commercial companies will do harm to the open source software and its community .

References

Bates, J., Di Bona, C., Lakhani, K. & Wolf, B. (2002), 'The boston consulting group hacker survey', http://129.3.20.41/eps/io/papers/0502/0502011.pdf, accessed on March 19th, 2009.

Bessen, J. (2004), 'Open source software: Free provision of complex public goods', http://www.researchoninnovation.org/opensrc.pdf, accessed on July 19th, 2005.

Bobko, P. K. (2000), 'Linux and general public licenses: Can copyright keep open source software free?', *AIPLA Quarterly Journal* **28 (1)**, 81–XX.

Bonaccorsi, A. & Rossi, C. (2003*a*), 'Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business', SSRN working paper.

Bonaccorsi, A. & Rossi, C. (2003*b*), 'Licensing schemes in the production and distribution of open source software: An empirical investigation', SSRN working paper.

Bonaccorsi, A. & Rossi, C. (2005), 'Why profit-oriented companies enter the OS field?: Intrinsic vs. extrinsic incentives', *ACM SIGSOFT Software Engineering Notes* **30(4)**, 1–5.

Dalle, J. M. & David, P. A. (2003), 'The allocation of software development resources in "open source" production mode', Working Paper, http://129.3.20.41/eps/io/papers/0502/0502011.pdf, accessed on March 19th, 2009.

Feller, J. & Fitzgerald, B. (2002), *Understanding Open Source Software Development*, Addison Wesley, Boston, MA.

Fink, M. (2003), *The Business and Economics of Linux and Open Source*, Person Education, Prentice Hall, NJ.

Gallini, N. T. (1984), 'Deterrence by market sharing: A strategic incentive for licensing', *American Economic Review* **74**, 931–941.

Hars, A. & Ou, S. (2002), 'Working for free? motivations for participating in open-source projects', *International Journal of Electronic Commerce* **6**, 25–39.

Hawkings, R. E. (2004), 'The economics of the open source software for a competitive firm', *NETNOMICS: Economic Research and Electronic Networking* **6**, 103–117.

Hecker, F. (1999), 'Setting up shop: The business of open-source software', *IEEE Software* **16(1)**, 45–51.

Hertel, G., Niedner, S. & Herrmann, S. (2003), 'Motivation of software developers in open source projects: An internet-based survey of contributors to the linux kernel', *Research Policy* **32**, 1159–1177.

Horne, N. T. (2001), 'Open source software licensing: Using copyright law to encourage free use', *Georgia State University Law Review* **17**.

Katz, M. L. & Shapiro, C. (1986), 'How to license intangible property', *Quarterly Journal of Economics* **101**, 567–589.

Kuan, J. (2001), 'Open source software as consumer integration into production', Working Paper, Stanford University.

Lakhani, K. & Von Hippel, E. (2003), 'How open source software works: Free user-to-user assitance', *Research Policy* **32**, 923–943.

Lee, S. H. (1999), 'Open source software licensing', unpublished working paper, Harvard University.

Lerner, J. & Tirole, J. (2002), 'Some simple economics of open source', *Journal of Industrial Economics* **50**, 197–234.

Lerner, J. & Tirole, J. (2004), 'The economics of technology sharing: Open source and beyond', NBER Working Paper No. 10956.

Raymond, E. S. (2000*a*), 'The cathedral and the bazaar', http://www.catb.org/esr/writings/cathedralbazaar/cathedral-bazaar, accessed on July 19th, 2005.

Raymond, E. S. (2000*b*), 'Homesteading the noosphere', http://www.catb.org/esr/writings/cathedralbazaar/homesteading, accessed on July 19th, 2005.

Rockett, K. E. (1990), 'Choosing the competition and patent licensing', *Rand Journal of Economics* **21**, 161–172.

Stallma, R. (1984), 'The GNU manifesto', http://www.gnu.org/gnu/manifesto.html, accessed on July 19th, 2005.

Tirole, J. (1997), *The Theory of Industrial Organization*, MIT Press, Cambridge, MA.

Torvalds, L. & Diamond, D. (2002), *Just for Fun: The Story of an Accidental Revolutionary*, Harper Collins Publishers, New York, NY.

Valimaki, M. (2005), *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry*, Turre Publishing.

CHAPTER 2

Open Source: A Strategic Approach

Open Source: A Strategic Approach

Yu Wang*
University of Aarhus

May 10, 2009

Abstract

This paper explores strategic reasons for profit-pursuing firms to adopt open source method, that is, give away the source codes of their software products and how open source software impacts on software industry. By setting up a two-stage competition game based on a Hotelling linear city model, I demonstrate that when heterogeneous consumers need to buy software and enhancement in two stages and consumers' valuation on enhancement is dependent on the adoption base of software because of the network effect, firms may have incentive to invite the dominance of open source software but compete on enhancement market, which will relax overall competition, as long as the network effect is low and enhancement is more differentiated than software. In that case, open source software increases the industry's profitability and each firm is better off. However, proprietary software can still dominate the market when network effect is high.

JEL classifications: L11, L13, L17, L86, D45

Keywords: Hotelling linear city, open source, network effect, product differentiation

*Address correspondence: Yu Wang, Department of Management, Building 322, University of Aarhus, DK-8000 Aarhus C, Denmark. E-mail: ywang@econ.au.dk

1 Introduction

Two decades ago, the concept of open source may be known only to those programming engineers who happened to be familiar with the GNU Project launched by Richard Stallman to create a free Unix-like operating system. One decade ago, open source became popular among the programmers thanks to the success of Linux. The popularity does not change the fact that open source is still regarded as an alternative software development method, which is characterized by free access of source codes and many voluntary developers working for free. Very few people expect it to bring a business revolution. However, the success of open source software in recent years has completely changed it. Here are some astonishing facts. More than two-thirds of websites are hosted using Apache, an open-source product that vanquishes commercial rivals. Its users include Google, Amazon and eBay, etc. Wikipedia, an open source online encyclopedia, attracts 683 million visitors annually reading over 10 million articles in 253 languages, comprising a combined total of over 1.74 billion words.[1] Many companies now see the great business potential through the success of open source software and begin to cooperate with open source projects by donating a huge amount of money. A good example is the donation from IBM to the Linux and Apache. Some companies also launched their own open source projects, such as Java and OpenOffice from SUN. Some companies even go further by totally building their business on the open source software, such as RedHat, a major Linux distribution vendor. It was founded in 1993 and went to public in 1999. The revenue from subscription, training and service in 2007 was about 400 million dollars.

Why are the profit-pursuing firms willing to give away the source code, which is regarded as the top secret and the profit origin? Many research works have been done to study the motivations of profit-pursuing firms to build up their business based on open source software. The possible reasons include: revenues by selling related products or software-related services [Wichmann (2002)]; exploitation of the R&D activity from the developers [Lakhani & Von Hippel (2003)]; Bugs fixing by the community [Von Hippel (2003)]; high quality and security concerns [Fink (2003)]. However, disclosure of source code may also arise for strategic reasons.

[1] According to the statistics from the official website of Wikepedia. See *http://en.wikipedia.org/wiki/Wikipedia* for more details.

Software products normally are regarded as durable goods. In its life span, the users probably need to buy the enhancements to work with the original software products to obtain more utility. These enhancements may include updates, patches or complementary software products. Thanks to the technical barrier and intellectual property protection, software users normally can only purchase enhancements from the same software distributor. However, when a software distributor chooses open source strategy, this monopoly power will be changed. According to the definition from Open Source Initiative, as long as the programme with *source code* can be freely accessed, modified and redistributed, loosely speaking, it is called open source software.[2] Open source software entirely eliminates the technical and legal barrier which prevents the rivals from providing enhancements to its users. Competition in the enhancement market will be reinforced. On the other hand, free distribution of open source software products will impose great competition pressure on the rival like an aggressive pricing strategy, which may increase the adoption base of open source software.

This paper sets up a two-period Hotelling model [see Hotelling (1929) for details] where two for-profit firms compete against each other for the consumers with heterogeneous preference on software and its enhancement in two periods respectively. At the same time, the adoption base of software has a positive network effect on the valuation of consumers on enhancement. Both firms can choose either open source or proprietary strategy. Once open source strategy is chosen, software will be distributed at zero price and there is no switching costs at all for the software users purchase enhancement from the rival. On the contrary, firms with proprietary strategy has the freedom to set the price of software and the full monopoly power over the provision of enhancement to its users. In other words, the huge switching costs will force the software users to purchase enhancement from the same firm. I find that open source strategy may soften overall competition to make the firms to obtain more profits when product differentiation degree of enhancement is higher than software and network effect is low. At this time, open source software will dominate the market and both firms are better off. But this dominance can never happen as long as the product differentiation degree of enhancement is lower than software. I also find that

[2]For details of definition of open source software, please visit Open Source Initiative (http://www.opensource.org)

when the network effect is high, the unique equilibrium will be that both firms choose proprietary strategy and evenly split the market.

This paper is related to the literature on switching costs and competing networks. Switching costs has been extensively studied in the last two decades, and the overview and survey of early research works can be found in Klemperer (1995) and Beggs & Klemperer (1992). Marinoso (2001) studies endogenous switching costs and product compatibility. He concludes that producers of product systems, which consist of different components, can have strong incentives to achieve compatibility. But this result heavily depends on the assumptions of homogeneity of components and low costs of reaching compatibility. Bouckaert & Degryse (2004) shows that in the two-period model, banks have incentives to disclose the borrower information to increase the rival's second-priod profits and decrease competition in the first-period market. Thus the overall competition is relaxed.

My paper resembles the last two papers to the extent that firms may have incentives to reduce the ex-post lock-in effect to invite entry in late periods to reduce the competition for the consumers in early periods, which can relax overall competition to generate more profits. The difference between my paper and theirs is that this paper deals with consumers with heterogeneous preference for both software products and enhancements. The difference between the degree of product differentiation in two stages does affect the choice of firms on whether to adopt open source strategy. And the open strategy may reinforce overall competition in some situations.

Another interesting paper by Shy & Thisse (1999) contains an similar idea to this paper. In their model, firms can choose either to protect their software (equivalent to a proprietary strategy in this paper) or not to protect it (equivalent to a open source strategy). There are two types of consumers: high valuation consumers who prefer the legal use and low valuation consumers who is indifferent between legal and illegal use. All consumers benefit from a positive network effect when the adoption base is increased. They show that there exists an equilibrium where both firms choose to drop software protection. They also find that this occurs when the network is strong. Interestingly, this paper finds a different conclusion: when network is strong, both firms will choose the proprietary strategy (equivalent to protect software in their paper). This difference arises because the different assumption of object of network effect. In Shy and Thisse's paper, users have higher valuation on software when it is adopted by larger population. However, my paper assumes that larger adoption base will only lead to higher valuation of enhancement instead of software itself because of the asymmetric information between firms and consumers.

The remainder of the paper is organized as follows. Section 2 presents the model and Section 3 offers the conclusion and the further discussion of the paper. Appendix contains the details of the solutions in the model.

2 Model

Consider a market where a group of heterogeneous users uniformly distributed from $[0,1]$. Firm A and B are located at 0 and 1 respectively. Firm A is the incumbent and Firm is the entrant. Each firm has two available strategies: proprietary and open source. Once proprietary strategy is chosen, the source code is closed and protected from each other. But if open source strategy is adopted, its source code will be disclosed to public.

Stage I: Firm A and B sequentially choose their strategy first. After each firm observes the rival's move, they simultaneously set the price of their software product, p_{AS} and p_{BS}.

Assumption 1: The development cost for each firm is equal to 0.

This assumption is to exclude the influence of development cost, so the study can focus on the strategic effect of open source strategy.

For user x where $x \in [0,1]$, her utility is

$$U_x^1 = \begin{cases} \beta - rx - p_{AS} \text{ if she buys software from Firm A} \\ \beta - r(1-x) - p_{BS} \text{ if she buys software from Firm B} \\ 0 \text{ if she does not buy software} \end{cases}$$

If the user chooses software from Firm i where $i = A, B$, she is called an i-user. Here r is the degree of production differentiation of software in Stage I. To simplify the analysis without losing generality, I assume $r = 1$ from now on in this paper.

Stage II: After buying software from firm i, the i-user need to buy the complementary enhancement package to work with the original software. This package will enrich the functionality of software they bought, which could be complementary accessories, extension applications or bug-fixing patches.

2.1 Two Proprietary Competitors

Now I suppose both of firms choose proprietary strategy.

Assumption 2: The Firm i can not provide the complementary enhancement package to j-user in Stage II if $i \neq j$.

When firms do not disclose source code of software, it is difficult for one firm to provide the complementary enhancement package which can work perfectly with software produced by the competitor because of the technical barriers. Or overcoming the technical problems will incur high cost. This assumption just rules out the possibility of compatibility Then the utility for A-user y is

$$U_{yA}^2 = \begin{cases} \delta n_{AS} - ty - p_{AE} \text{ if she buys enhancement from Firm A} \\ 0 \text{ if she doesn't buy enhancement from Firm A} \end{cases} \tag{1}$$

where $y \in [0, n_{AS}]$, n_{AS} is the population of users who bought software from Firm A at Stage I and p_{AE} is the price of the complementary enhancement package sold by Firm A. Similarly, $t > 0$ is the degree of production differentiation of enhancement in Stage II. Here $\delta > 0$ is the network effect factor.

Similarly, the utility for B-user z is

$$U_{zB}^2 = \begin{cases} \delta n_{BS} - t(1-z) - p_{BE} \text{ if she buys enhancement from Firm B} \\ 0 \text{ if she doesn't buy enhancement from Firm B} \end{cases}$$

where $z \in [0, n_{BS}]$, n_{BS} is the population of users who bought software from Firm B at Stage I and p_{BE} is the price of the complementary enhancement package sold by Firm B.

There is no cost for providing the complementary enhancement package.

Assumption 3: Each user only knows his own preference and only firms know the distribution of users' preference. The adoption base of software will be common knowledge in stage II.

This assumption rules out the possibility that users can obtain negative utility in Stage I to maximize the total utility from two stages. This assumption is quite close to the reality in the software market where the consumers and firms have asymmetric information.

Assumption 4: Enhancement is always more differentiated than software, that is, $t \geq 1$.

Software can be regarded as a set of functions, by which users can achieve some specific tasks. And enhancement either improves the existing functions or adds new ones. When the consumers make their purchase decision, they will probably to take all functions into consideration. Some functions may be quite useful to them, but some may not. Normally users' preference over all functions as a whole should be less heterogeneous than some specific functions. So enhancement should be no less differentiated than software.

Lemma 1 *Let p_{AS} and p_{BS} be any pair of non-negative prices. If $\beta \geq \frac{3}{2}$, there is a unique and stable adoption equilibrium that all users buy software.*

Proof. If user x is indifferent between buying software from Firm A and B, then x must satisfy

$$\beta - x - p_{AS} = \beta - (1 - x) - p_{BS}$$

then

$$x = \frac{p_{BS} - p_{AS} + 1}{2}$$

Firm A will choose p_{AS} to maximize the profits

$$\pi_{AS} = p_{AS}x$$

Then Firm A will always set

$$p^*_{AS} = \frac{p_{BS} + 1}{2}$$

Since Firm A and Firm B are symmetric,

$$p^*_{AS} = p^*_{BS} = 1$$

and $x = 1$. If all users buy software, β must satisfy

$$\beta - x - p_{AS} \geq 0$$

So $\beta \geq \frac{3}{2}$ ■

In this paper, I will focus on the case where all users buy software in the first stage. That is, I assume $\beta \geq \frac{3}{2}$ in the rest of the paper. Similarly, I also assume that $\delta > \frac{3}{2}t$ in the rest of the paper.

I use the method of backward induction to solve for the equilibrium.

In Stage II, since both firms choose proprietary strategy, each of them will charge its software users a monopoly price on enhancement. Firm A faces n_{AS} users who buy software from him in Stage I and will set price p_{AE} to maximize

$$\pi_{AE} = p_{AE} n_{AE} \tag{2}$$

where n_{AE} is the number of users who purchase the complementary enhancement package from Firm A and $n_{AE} \leq n_{AS}$. Combine 1 and 2, the solutions of the maximum problem can be easily obtained as follows:

$$p^*_{AE} = \begin{cases} \frac{\delta}{2} n_{AS} \text{ if } \frac{3}{2}t \leq \delta < 2t \\ (\delta - t) n_{AS} \text{ if } \delta \geq 2t \end{cases}$$

$$n^*_{AE} = \begin{cases} \frac{\delta}{2t} n_{AS} \text{ if } \frac{3}{2}t \leq \delta < 2t \\ n_{AS} \text{ if } \delta \geq 2t \end{cases}$$

$$\pi^*_{AE} = \begin{cases} \frac{1}{4t}\delta^2 n^2_{AS} \text{ if } \frac{3}{2}t \leq \delta < 2t \\ (\delta - t) n^2_{AS} \text{ if } \delta \geq 2t \end{cases} \tag{3}$$

Now back to Stage I, Firm A will choose p_{AS} to maximize the total profit

$$\begin{aligned} \pi_A &= \pi_{AS} + \pi^*_{AE} \\ &= p_{AS} n_{AS} + \pi^*_{AE} \end{aligned}$$

where

$$n_{AS} = \frac{p_{BS} - p_{AS} + 1}{2}$$

and π^*_{AE} is defined as (3).

The detailed solution can be found in Appendix A.

After reviewing the solution, the following proposition can be easily reached.

Proposition 1 *If both firms compete against each with proprietary strategy, they* *will choose the same price and evenly split the market in both stages for software and enhancement respectively in the equilibrium.*

1) Equilibrium price of software always decreases with network effect δ;

2) Equilibrium price of software is positive and increases with production differentiation degree of enhancement t when $1 < t < \frac{16}{9}$ and $\frac{3}{2}t < \delta \leq 2\sqrt{t}$; otherwise equilibrium price of software is always equal to zero;

When both firms choose proprietary strategy, the competition is concentrated in Stage I and both of them are monopolistic providers of enhancement to their software users in Stage II. And the intensity of competition is greatly influenced by monopoly profits firms can obtain from enhancement in Stage II. When network effect is greater, each firm can obtain more profits if the adoption base of its software is increased, which will induce them to compete fiercer in Stage I. The interesting finding here is how the production differentiation degree of enhancement changes the competition level. When differentiation of enhancement is slightly higher than software ($1 < t < \frac{16}{9}$) and network effect is low, both firms will set software price strictly positive and it increases with degree of enhancement differentiation. In other words, higher differentiation of enhancement will soften the competition as long as it is not too high. But if it is greater than some level ($t \geq \frac{16}{9}$), software price lowers to zero and the competition is maximized. The explanation of this finding is that when network effect is low and enhancement differentiation is increased but lower than some degree, two firms can always get more profits if they soften the competition in the first stage. Because the increase of adoption base by aggressive pricing will not bring more profits from the enhancement than the loss of profits from software due to the lower price. On the other hand, when enhancement differentiation exceeds some degree, each has strong incentive to lower the software price to expand the adoption base of its software users, which will bring more profits from enhancement.

2.2 Proprietary Incumbent versus Open Source Entrant

Now I suppose that the incumbent is bound to proprietary strategy but the entrant adopts open source strategy. From now on, Assumption 2 no longer holds. Since there is no cost to produce enhancements, Firm A will always provide two versions of the complementary enhancement package, one for its own software and the other for the competitor's software if its rival chooses open source strategy.

The B-user w's utility in Stage II will be

$$U^O_{wB} = \begin{cases} \delta n_{BS} - tw - p_{AEB} \text{ if she buys enhancement from Firm A} \\ \delta n_{BS} - t(1-w) - p_{BEB} \text{ if she buy enhancement from Firm B} \\ 0 \text{ if she does not buy enhancement} \end{cases}$$

where n_{BS} is the number of users who purchased software from Firm B and p_{AEB} is the price of enhancement provided to B-users from Firm A and p_{BEB} is the price of enhancement provided to B-users from Firm B. Here $w \in [0, n_{BS}]$

In Stage II, Firm A will provide two versions of enhancement since Firm B chooses open source strategy. But Firm B can only provide one version of enhancement to B-users. For B-user v, she will be indifferent between enhancement provided by either Firm A or B if

$$\delta n_{BS} - v - p_{AEB} = \delta n_{BS} - (n_{BS} - v) - p_{BEB}$$

where $v \in [0, n_{BS}]$. Since $\delta \geq \frac{3}{2}t$, all B-users will buy enhancement either from Firm A or B. After choosing open source strategy, Firm B will obtain zero profit from Stage I because $p_{BS} = 0$. His optimal problem is just to set p_{BEB} to maximize $p_{BEB}n_{BEB}$, where n_{BEB} is the number of B-users who purchase enhancement from Firm B. Firm A in Stage II will maximize

$$\begin{aligned} \pi_{AE} &= \pi_{AEA} + \pi_{AEB} \\ &= p_{AEA}n_{AEA} + p_{AEB}n_{AEB} \end{aligned}$$

by setting p_{AEA} and p_{AEB}. Here n_{AEA} and n_{AEB} are the numbers of A-users and B-users who buy enhancement from Firm A respectively.

The solutions are

$$\begin{cases} p^*_{AEB} = p^*_{BEB} = (1 - n_{AS})t \\ n^*_{AEB} = n^*_{BEB} = \frac{1}{2}(1 - n_{AS}) \end{cases}$$

and

$$\begin{cases} p^*_{AEA} = \frac{1}{2}\delta n_{AS} \\ n^*_{AEA} = \frac{1}{2t}\delta n_{AS} \end{cases} \text{when } \frac{3}{2}t \leq \delta < 2t$$

$$\begin{cases} p^*_{AEA} = (\delta - t)n_{AS} \\ n^*_{AEA} = \delta n_{AS} \end{cases} \text{when } \delta \geq 2t$$

In stage I, Firm A will set price p_{AS} to maximize

$$\pi_A = \pi_{AE} + \pi_{AS}$$

where $\pi_{AS} = p_{AS}n_{AS}$, $n_{AS} = (\frac{1-p_{AS}}{2})$ and $0 \leq p_{AS} \leq 1$.
The details of solutions can be found in Appendix B.

Proposition 2 *When the incumbent chooses proprietary strategy but the entrant chooses open source strategy, then*

1) if $\delta \geq \frac{5}{2}t$, open source software and proprietary software will coexist in the market. But open source software will drive proprietary software out of the market as long as $\frac{3}{2}t \leq \delta < \frac{5}{2}t$;

2) the incumbent will always enjoy equal or higher profits than the entrant;

Proof. 1) When $\frac{3}{2}t \leq \delta < \frac{5}{2}t$, $p_{AS} = 1$ and $p_{BS} = 0$, which means $n_{AS} = 0$ and $n_{BS} = 1$. When $\delta \geq \frac{5}{2}t$, $p_{AS} = p_{BS} = 0$ and $n_{AS} = n_{BS} = \frac{1}{2}$

2) $p_{AEB} = p_{BEB}$ and $n_{AEB} = n_{BEB}$, so $\pi_B = p_{BEB}n_{BEB} = p_{AEB}n_{AEB} = \pi_{AEB} \leq \pi_{AEA} + \pi_{AEB} + \pi_{AS} = \pi_A$. ■

When the entrant chooses open source strategy to compete against the incumbent with proprietary strategy, he can always easily obtain more software market because of the low price. On the other hand, the entrant's total profits will be no more than the incumbent's because he sacrifices all potential profits from software markets and gives up partial profits from his users who buy enhancement from his rival. So we can predict that in a market where an open source software firm compete with a proprietary software firm in two periods for both software and enhancement, the proprietary software company will normally enjoy the higher profits than the open source company. So even open source software may help the entrant penetrate the market and obtain more market share, sometimes its presence may also benefit the proprietary incumbent.

The proprietary incumbent's competition strategy is dependent on both the network effect and enhancement differentiation. When enhancement is more differentiated than software, the incumbent will always choose to compete in the software market as long as the network effect is high. High network effect gives the incumbent strong incentive to enlarge the adoption base of its software in the first stage to earn high monopoly profits later. But if network effect is not high, the incumbent is willing to give up the whole software market and compete with its rival on enhancement in the second stage.

2.3 Two Open Source Competitors

Now I suppose that both firms choose open source strategy.

In stage I, both firms are forced to set prices equal to zero and evenly split the market in the first stage, that is,

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \end{cases}$$

Since both choose open source strategy, users can buy enhancement from either of firms. Then i-user w's utility in Stage II will be

$$U^O_{wi} = \begin{cases} \delta n_{iS} - tw - p_{A\,\mathrm{Ei}} \text{ if she buys enhancement from Firm A} \\ \delta n_{iS} - t(1-w) - p_{B\,\mathrm{Ei}} \text{ if she buy enhancement from Firm B} \\ 0 \text{ if she does not buy enhancement} \end{cases}$$

where n_{iS} is the number of users who purchased software from Firm i and $p_{A\,\mathrm{Ei}}$ is the price of enhancement provided to i-users from Firm A and $p_{B\,\mathrm{Ei}}$ is the price of enhancement provided to i-users from Firm B. Here $w \in [0, n_{iS}]$.

In the second stage, both firms will provide two versions of enhancement to the market. The enhancement prices and number of user buying enhancement from each firm are

$$\begin{cases} p^*_{AEA} = p^*_{AEB} = p^*_{BEA} = p^*_{BEB} = \frac{1}{2} \\ n^*_{AEA} = n^*_{AEB} = n^*_{BEA} = n^*_{BEB} = \frac{1}{4} \end{cases}$$

So the profit of each firm is

$$\pi_A^* = \pi_B^* = \frac{1}{4}$$

Proposition 3 *As long as enhancement is more differentiated than software, it is impossible for both firms to choose open source strategy.*

Proof. Suppose Firm B chooses open source strategy. If Firm A chooses open source strategy as well, then $\pi_A^{OO} = \pi_B^{OO} = \frac{1}{4}$. However, if Firm A chooses proprietary strategy and $t \geq 1$, his profits will be

$$\pi_A^{PO} = \begin{cases} \frac{t}{2} \text{ if } \frac{3}{2}t \leq \delta < \frac{5}{2}t \\ \frac{2\delta - t}{8} \text{ if } \delta \geq \frac{5}{2}t \end{cases}$$

It is obvious that $\pi_A^{PO} \geq \pi_A^{OO}$. So it is always for Firm A to choose proprietary strategy when Firm B chooses open source strategy. ■

At the same time, since the incumbent always chooses his strategy first, which give him a strategic advantage. Combining Proposition 2 and 3 leads to the following corollary.

Corollary 1 *It is always optimal for the incumbent to adopt proprietary strategy.*

Now let's analyze the equilibrium of the model.

Proposition 4 *When enhancement is more differentiated than software, there will be one unique Nash Equilibrium that the incumbent choose proprietary strategy but the entrant chooses open source strategy as long as $\frac{3}{2}t \leq \delta < \frac{5}{2}t$; the unique Nash Equilibrium is that both firms choose proprietary strategy when $\delta \geq \frac{5}{2}t$.*

Proof. The entrant's profit under open source strategy is

$$\pi_B^{PO} = \begin{cases} \frac{t}{2}, \text{ when } \frac{3}{2}t \leq \delta < 2t \text{ and } t \geq 1 \\ \frac{t}{8}, \text{ when } \delta \geq \frac{5}{2}t \text{ and } t \geq 1 \end{cases}$$

and the profit under proprietary strategy is

$$\pi_B^{PP} = \begin{cases} \frac{8t-\delta^2}{16t}, \text{ when } \frac{3}{2}t \leq \delta < 2\sqrt{t} \text{ and } 1 \leq t < \frac{16}{9} \\ \frac{\delta^2}{16t}, \text{ when } 2\sqrt{t} \leq \delta < 2t \text{ and } 1 \leq t < \frac{16}{9} \\ \frac{\delta-t}{4}, \text{ when } \delta \geq 2t \text{ and } 1 \leq t < \frac{16}{9} \\ \frac{\delta^2}{16t}, \text{ when } \frac{3}{2}t \leq \delta < 2t \text{ and } t \geq \frac{16}{9} \\ \frac{\delta-t}{4}, \text{ when } \delta \geq 2t \text{ and } t \geq \frac{16}{9} \end{cases}$$

so when $t \geq 1$,

$$\begin{cases} \pi_B^{PO} - \pi_B^{PP} \geq 0 \text{ if } \frac{3}{2}t \leq \delta < \frac{5}{2}t \\ \pi_B^{PO} - \pi_B^{PP} \leq 0 \text{ if } \delta \geq \frac{5}{2}t \end{cases}$$

■

From the propositions above, an interesting conclusion can be reached that as long as the enhancement is more differentiated than software and network effect is not high, the proprietary incumbent is willing to drop the competition on software in the first stage to let the open source entrant take away the whole market share. By doing this, he can earn more profits from the competition on enhancement in the second stage. In the case where $\frac{3}{2}t \leq \delta < \frac{5}{2}t$ and $t \geq 1$, all the competition is concentrated in the first stage where software differentiation degree is equal to 1 before open source strategy is available to the entrant. After the adoption of open source strategy by the entrant, all competition is transferred from the first stage to the second stage. To be specific, competition is shifted from low differentiation market to high differentiation market, which will definitely dampen the overall competition. The following corollary can be easily obtained from Proposition 4.

Corollary 2 *When* $\frac{3}{2}t \leq \delta < \frac{5}{2}t$ *and* $t \geq 1$, *open source software make each firm better off.*

Proof. $\pi_A^{PO} = \pi_B^{PO} \geq \pi_B^{PP} = \pi_A^{PP}$ if $\frac{3}{2}t \leq \delta < \frac{5}{2}t$ and $t \geq 1$. So the profits of both the proprietary incumbent and the open source entrant are greater than when both of them choose proprietary strategy. ■

Another interesting finding is worth addressing here. Proposition 4 suggests that the firms are in "Prisoners' Dilemma" situation. When the network effect is so strong to dominate product differentiation and demand consumer coordination, both firms follow proprietary strategy that result in a fragmented market.

The explanation of this phenomenon is the strong network effect can dramatically increase the users' valuation on enhancement thus they are willing to pay more. But this high valuation will also lead to strong incentive of competition in the market of enhancement. So compared with competing in a uniform market, monopoly in a fragmented market is more profitable regardless of product differentiation.

3 Conclusion

This paper builds a model to study the strategic reasons for profit-pursing firms choosing open source method and impact of open source software on the software industries. When consumers need to purchase both software and enhancement sequentially, traditional software distributors will compete intensively for software users to create ex-post lock-in, thus earn monopolistic profits from providing enhancement. This paper shows that software distributors may intentionally invite the dominance of open source software and compete on the market of enhancement, thus to relax overall competition when the network effect is low and enhancement is more differentiated than software. Though both the proprietary incumbent and the open source entrant earn zero profit from software market if the dominance of open source software is invited, they can obtain more profits from the enhancement market where the software users have higher willing to pay because of the uniform software market.

This paper contributes to the existing literature in two directions. First, open source software will not necessarily increase the competition and lower the profits of software industry as many people think because it is distributed for free. On the contrary, software companies may have incentive to embrace open source strategy to relax the intensive competition in software market but obtain high profit from providing enhancement later. In this situation, the presence of open source software will not reduce the profitability of software market. Instead, each competing firm is better off compared with the situation when both of them choose to close the source code. This conclusion helps to understand why so many companies now adopt open source strategy, including the giant incumbents like IBM and SUN.

Second, this paper also finds how the production differentiation and network effect influence the success of open source software. The imbalance development of open source software is a quite interesting phenomenon in software market. Some open source software can easily dominate the market like Apache. But some of them coexist with its proprietary rival like Linux and some are even in danger of extinction. This paper can shed some new light on the explanation. For example, Linux's performance is quite different in server market and desktop market. According to IDC's report for Q1 2007, Linux now holds 12.7% of the overall server market. But for the desktop market share, Linux is estimated range from 1%-2%. In comparison, Microsoft operating systems hold more than 90%. Mac system accounts for the rest market share.[3] So we can see Linux is almost driven out of desktop market. Of course, the high requirement of computer skills and bad interface are two reasons for its bad performance. However, the high network effect in desktop market also plays an important role. High market share will lead to high valuation of software products in Windows platform. This also explains why Microsoft competes extremely intensively with open source software, such as promotions all over the world and tolerance of piracy in developing countries. The success of Apache and the difficulty faced by OpenOffice can also be explained by the model in my paper.

At last, there are some remarks of the model worth further discussion. This paper assumes the zero development cost to focus on studying strategic reasons of firms adopting open source strategy. However, development cost is also a key reason for open source's popularity. When firms consider whether to enter some specific market, high development cost and intensive competition probably make both firms stay out of the market even though it is profitable for either of them to serve the market alone. Or sometimes the incumbent can easily deter the entry of proprietary entrant. In this case, there may exist a unique equilibrium where both firms choose open source strategy and open source software has more possibility to dominate the market. Second, this paper rules out the possibility that software is more differentiated than enhancement. If it does happen, then open source software can never drive proprietary software out the market and

[3] This data is from Wikipedia, for details please check http://en.wikipedia.org/wiki/Linux

coexisting is the best results.

References

Beggs, A. & Klemperer, P. (1992), 'Multi-period competition with switching costs', *Econometrica* **60(3)**, 651–666.

Bouckaert, J. & Degryse, H. (2004), 'Softening competition by inducing switching in credit markets', *Journal of Industrial Economics* **52**, 27–52.

Fink, M. (2003), *The Business and Economics of Linux and Open Source*, Person Education, Prentice Hall, NJ.

Hotelling, H. (1929), 'Stability in competition', *The Economic Journal* **39, No. 153**, 41–57.

Klemperer, P. (1995), 'Competition when consumers have switching costs: An overview with applications to industrial organization, macroeconomics, and international trade', *Reveiw of Economics Studies* **62**, 515–539.

Lakhani, K. & Von Hippel, E. (2003), 'How open source software works: Free user-to-user assitance', *Research Policy* **32**, 923–943.

Marinoso, B. G. (2001), 'Technological incompatibility, endogenous switching costs and lock-in', *Journal of Industrial Economics* **49**, 281–298.

Shy, O. (1998), *Industrial Organization: Theory and Applications*, MIT Press, Cambridge, Massachusetts.

Shy, O. (2001), *The Economics of Network Industries*, Cambridge University Press.

Shy, O. & Thisse, J. F. (1999), 'A strategic approach to software protection', *Journal of Economics and Management Strategy* **8**, 163–190.

Von Hippel, E. (2003), 'Open source projects as horizontal innovation networks - by and for users', MIT Sloan School of Management Working Paper 4366-02.

Wichmann, T. (2002), 'Firms' open source activities: Motivations and policy implications', Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Appendix A

When both firms choose proprietary strategy, Firm A's maximum problem can be formalized as follow:

$$\max \pi_A = p_{AS}\frac{p_{BS}-p_{AS}+1}{2} + \pi^*_{AE}, \text{ where } \pi^*_{AE} = \begin{cases} \frac{1}{4t}\delta^2 n^2_{AS} \text{ if } \frac{3}{2}t \le \delta < 2t \\ (\delta - t)n^2_{AS} \text{ if } \delta \ge 2t \end{cases}$$

$$s.t. \begin{cases} 0 \le p_{AS} \le 1 \\ 0 \le p_{BS} \le 1 \\ t > 1 \end{cases}$$

By solving this and because of the symmetry of Firm A and Firm B, the following results can be reached:

When $1 \le t < \frac{16}{9}$,

1) if $\frac{3}{2}t \le \delta < 2\sqrt{t}$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 1 - \frac{1}{4t}\delta^2 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \\ p^*_{AE} = p^*_{BE} = \frac{\delta}{4} \\ n^*_{AE} = n^*_{BE} = \frac{\delta}{4t} \\ \pi^*_A = \pi^*_B = \frac{1}{2} - \frac{1}{16t}\delta^2 \end{cases}$$

2) if $2\sqrt{t} \le \delta < 2t$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \\ p^*_{AE} = p^*_{BE} = \frac{\delta}{4} \\ n^*_{AE} = n^*_{BE} = \frac{\delta}{4t} \\ \pi^*_A = \pi^*_B = \frac{1}{16t}\delta^2 \end{cases}$$

3) if $\delta \ge 2t$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \\ p^*_{AE} = p^*_{BE} = \frac{1}{2}(\delta - t) \\ n^*_{AE} = n^*_{BE} = \frac{1}{2} \\ \pi^*_A = \pi^*_B = \frac{1}{4}(\delta - t) \end{cases}$$

When $t \ge \frac{16}{9}$,

1) if $\frac{3}{2}t \leq \delta < 2t$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \\ p^*_{AE} = p^*_{BE} = \frac{\delta}{4} \\ n^*_{AE} = n^*_{BE} = \frac{\delta}{4t} \\ \pi^*_A = \pi^*_B = \frac{1}{16t}\delta^2 \end{cases}$$

2) if $\delta \geq 2t$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ n^*_{AS} = n^*_{BS} = \frac{1}{2} \\ p^*_{AE} = p^*_{BE} = \frac{1}{2}(\delta - t) \\ n^*_{AE} = n^*_{BE} = \frac{1}{2} \\ \pi^*_A = \pi^*_B = \frac{1}{4}(\delta - t) \end{cases}$$

Appendix B

When Firm A chooses proprietary strategy but Firm B chooses open source strategy, the equilibrium prices and profits of each firm are the following:

When $t \geq 1$,

1) if $\frac{3}{2}t \leq \delta < \frac{5t}{2}$, then

$$\begin{cases} p^*_{AS} = 1 \\ p^*_{BS} = 0 \\ \pi^*_A = \pi^*_B = \frac{t}{2} \end{cases}$$

2) if $\delta \geq \frac{5t}{2}$, then

$$\begin{cases} p^*_{AS} = p^*_{BS} = 0 \\ \pi^*_A = \frac{2\delta - t}{8} \\ \pi^*_A = \pi^*_B = \frac{t}{8} \end{cases}$$

CHAPTER 3

Optimal Licensing Scheme: Proprietary, Open Source or Mixed?

Optimal Licensing Scheme: Proprietary, Open Source or Mixed?

Yu Wang*
University of Aarhus

May 10, 2009

Abstract

In this paper I present a theoretical model to study a for-profit firm's choice of optimal licensing scheme from three alternatives: pure proprietary, pure open source and mixed. The firm needs to consider how each licensing scheme influences the product differentiation and development costs when there exists network externality of software usage.

JEL classifications: L11, L17, L86, D45

Keywords: product differentiation, reciprocity terms, network externality

*Address correspondence: Yu Wang, Department of Management, Building 1322, University of Aarhus, DK-8000 Aarhus C, Denmark. E-mail: ywang@econ.au.dk

1 Introduction

According to the survey of web servers used by the million busiest websites in April 2009 by Netcraft, Apache is the champion with a 66% market share.[1] Its users include many famous websites you probably visit every day, such as Google, eBay and Amazon. Many Internet users now are quite familiar with another open source product Wikipedia, an online encyclopedia with around 2.5 million entries in more than 260 languages.[2] There are many other popular open source software products besides these two, which are widely adopted by consumers.

At the same time, the attitude to open source as a threat is also gradually changed in the business community. Some small start-up enterprises have seen the opportunity in open source and achieved great success, such as Red Hat and MySQL. Why do these firms build their business based upon open source software? Another interesting question will be how one firm chooses the optimal business model for itself from so many available choices on the market if he decides to adopt open source strategy? In other words, what is the optimal licensing scheme to distribute open source software?

In this paper, I set up a theoretical model where a profit-maximizing monopolist develops a software product targeted to a group of heterogeneous consumers. The firm can choose one of three alternative licensing schemes to develop and distribute the software: pure proprietary, pure open source or mixed. The firm's choice is dependent on which licensing scheme can bring the best product differentiation level to maximize the profit when software use exhibits positive network externality. The results suggest that firm are inclined to adopt the pure proprietary licensing scheme when the consumers are more support-independent. However, he may prefer to release the source code when the consumers are support-oriented. Specifically, the pure open source licensing scheme is dominant when the consumers are also insensitive to the reciprocity terms, otherwise the mixed strategy is preferred. At the same time, this paper also provides a possible explanation for diversification of open source licensing.

This paper relates to literature on economics of open source software in the following directions. First is the choice of open source license. Lerner & Ti-

[1] http://news.netcraft.com/archives/web_server_survey.html
[2] http://en.wikipedia.org/wiki/Wikipedia

role (2005) studies the determinants of the open source license choice. They presents an empirical analysis to explore the relationships between license choice and project characteristics. They argue that the licensor's choice will be affected by the preference of contributors besides his own, and both the licensor and contributors have a wide variety of motivations. In my paper, the firm also considers both how to use the licensing scheme to differentiate software product and the open source community's preference for his choice of licensing scheme.

Second is the relationship between firms and the open source communities. Dahlander & Magnusson (2005) proposes a typology of three different basic approaches used by firms to inter-relate to their communities: symbiotic, commensalistic and parasitic. Firms face different managerial issues and needs to use different operational means of subtle control to maintain a stable relation. Their another paper [Dahlander & Magnusson (2006)] also argues that the relations firms have with open source communities are important to create a sustainable business model. Firms need to balance between potential benefits from communities and space of firms' strategic actions after considering different interests and value of both firms and communities. Rossi & Bonaccorsi (2006) finds firms have high incentive to cater for the preference of open source community for receiving contributions and support from them and gaining competitive and cost advantages. In this paper, I also assume that the choice of different licensing scheme does influence the relation between firms and communities besides the product differentiation.

Third is competition between open source software and commercial software. Bitzer (2004) finds that when product heterogeneity is low the proprietary incumbent may be driven out of the market by the open source entrant because the competition makes the profits unable to cover the development cost. Casadesus-Masanell & Ghemawat (2006) sets up a dynamic mixed duopoly between Linux and Windows and Economidies & Katsamakas (2006) studies two-sided competition between proprietary and open source platform. Both of these two paper shows that network externality plays an important role for the adoption of open source software. However, this paper chooses an alternative approach to study how product differentiation and network externality influences the choice of open source in a monopoly instead of duopoly market. I finds that open source can also be used to increase monopoly profit.

The remainder of the paper is organized as follows. Section 2 is an overview of two popular open source business models. Section 3 develops an analysis of licensing schemes as versioning tools. Section 4 presents a basic model and Section 5 concludes.

2 Open Source Business Models

There are so many successful software companies building their business upon open source software. So what is open source software? Generally speaking, as long as a software product's programme is freely available to the public both in source code as well as compiled form, and allow the modification and free distribution of the original and derived works, it can be regarded as open source software. Once a software programme is distributed under an open source license, the licensor or the software owner just gives away the programme for free and also grant the modification and redistribution rights to the public. So it seems difficult to profit from open source software.

The motivations of individuals or organizations to be engaged in open source movement are quite complicated and diversified [see Hars & Ou (2002) and Von Krogh & Von Hippel (2006) for more details]. However, the reason of for-profit companies releasing their privately owned code to the public or setting up open source projects seems to be more obvious. Bonaccorsi & Rossi (2003) makes an empirical analysis to compare the motivations of individual programmers and firms to take part in the open source movement. They find that firms emphasize economic and technological reasons for adoption of open source strategy.

Traditional software companies generate profit by selling the license of the software product. To protect the intellectual property, the commercial software companies are reluctant to reveal the source code, which completely and clearly demonstrate the designing and programming process of software. However, software companies can enjoy several benefits if they disclose the source code. Firstly, a firm can greatly save research cost in software development by adopting open source strategy. Von Hippel (2003) regards exploiting the free contribution from open source communities, which makes the costly innovation more affordable, as one important reason of open source business. When source code is freely available, the programmers from open source communities can help to improve the quality of existing code and develop new features or functions. Most of all, all these works are voluntary and firms can get all these contributions from outside developers without paying anything. For those start-up software firms, the resource limitation and lack of funds make them difficult to produce high quality software and survive the fierce competition in software market. Open source strategy can help them overcome the cost disadvantage while sustain the high quality of software. Secondly, open source also brings firms strategic benefits by enlarging the adoption base. The developers from the open source community are also potential users. Early release of source code to them will induce them to adopt the software before it is finished. During the process of contributing to improve the quality of software, they may probably add some functions or features according to their own needs, which makes them loyal users when the software product is brought to the market. At the same time, the open source software can always be obtained at no or trivial cost, which is attractive to those users with high budget constraint. Both of these two characteristics will help open source software gain large population of users in quite a short time. When the usage of the software product exhibits network externalities, the software vendor can earn more profits from the increased willing to pay of the users for the same software without any quality improvement.

On the other hand, how could the software company generate actual revenue if he choose the open source strategy? Giving away the programme normally means the loss of profits from license fee. However, open source companies can use other clever ways to earn profits. Lerner & Tirole (2002) argues that companies can obtain indirect revenues by selling related products when they are engaged in open source projects. Feller & Fitzgerald (2002) address a new model of software as a consumer-driven service, that is, firms can profit from complementary service while giving away software for free.

I will present examples of two most popular open source business models today in the following to illustrate the process.

A. Red Hat: Support Seller

Red Hat, the world's leading open source and Linux provider was founded in 1994 and successfully had an IPO in 1999 after years of incredible growth. According to the survey by IDC in August 2001, Red Hat became the market leader for second consecutive year with 52.4% of the Linux shipments worldwide. Red Hat is a pure open source company which only distributes open source software, and code it writes is also licensed under the GNU General Public Licence (GPL).

Red Hat adopts a business model called support seller, which is the most common model in the Linux market. The basic principle of this model can be generalized as providing support to improve the user experience of Linux and profit from this service.

As we all know that Linux is developed by a large population of programmers all over the world in open source communities. Raymond (2000) uses the word *bazaar* to describe the development method of open source software. Modularity is widely used in the development of Linux to increase efficiency [Narduzzo & Rossi (2003)]. However, it may lead to an inherent problem: components are often separated from each. The users have to integrate these programmes themselves, which is quite inconvenient and time-consuming. Some unprofessional users are willing to pay for a package consolidating all these components and other necessary programmes. This is the first profiting method for Red Hat: integrating the components of Linux into a cohesive package and charge for the convenience. At the same time, Red Hat also generates revenue in the second way, that is, providing support and service to improve the user experience. When users adopt Linux, they can only get the necessary technical support from the other developers in open source communities. So the quality and timeliness are difficult to be guaranteed like the commercial software. Red Hat can provide the support to make installation, maintenance and updating much easier for the non-professional users.

In 2003, Red Hat split its Linux offerings into two products – the Fedora Project and Red Hat Enterprise Linux (RHEL). Fedora is an experimental version of Linux, and RHEL is a stable version, which is supported and certified by Red Hat. Of course, RHEL is not free. Most enterprise users are willing to choose RHEL while Fedora is the favorite of the professional developers who are good at programming and but sensitive to the price. At the same time, RHEL source code is available, but the compiled and ready-to-go code needed by computers to run the operating system is only available to those buying support subscription.

One fundamental problem of support seller model is that it may be unable to generate sustainable and enough profits in the long term. The open source firm choosing this model has no monopolistic control over the programme. Technically, anyone can also redistribute programme without any development cost. They can also provide the similar support to compete for the customers. Wichmann (2002) argues that the key of success of this business model is to increase differentiation by branding build. Actually Red Hat does put much effort into marketing in recent years. It raised sales and marketing expenses to $52.1 million during the fourth quarter of 2007.[3]

B. MySQL: Dual Licensing

MySQL is one of the most popular database systems which has more than 11 million installations[4]. Its competitors today includes Oracle Database, IBM DB2 and Microsoft SQL server. Most of all, it is also a open source software product. MySQL is owned by a single for-profit firm, MySQL AB, now a subsidiary of Sun Microsystems[5].

MySQL adopts a business model called dual licensing. It release its database software products under two licenses: open source and proprietary. Anyone can download the programme with the source code for free and redistribute it as long as the redistributed products are also under the open source license, specifically the GPL in MySQL's case. However, if you plan to distribute a new product integrated with the original code for commercial purpose, you have to choose the proprietary license which remove the GPL restrictions. Developers can always adopt, customize and integrate MySQL into their own larger software project. If he distribute the larger project also at the GPL, he does not need to pay any fee. However, if he plans to distribute it under the proprietary license, he must buy the proprietary license from the firm. And this two versions are totally identical except the license[6].

One important reason of dual licensing of MySQL is the strong network effect in the usage of database systems. The open source version is free and has the same code quality as the proprietary one, so it is easy to create a large population

[3] "Red Hat cheerleading purges Q4 profits", 27th March 2008, www.theregister.co.uk

[4] See www.mysql.com

[5] "Sun to fork out $1 billion for open-source firm MySQL", 16th January 2008, news.cnet.com

[6] Valimaki (2005) presents in details three well known examples of companies that have successfully adopted a dual licensing strategy, which include MySQL, Sleepycat, and Trolltech. MySQL is the most successful one among them.

of developers who are familiar with the product. When the commercial users are choosing from several alternative products, they are most likely to adopt one with largest user base because it is easier for them to find documentation, software add-ons and recruit trained developers in the market. So the dual licensing can certainly bring the firm more advantage in the competition besides providing a stable profiting method.

However, MySQL also has to meet some legal requirement and sacrifice some benefits from open source communities to implement the dual licensing model. First, MySQL must own and control the copyrights of software to obtain the legal right to distribute the software product under two different license. So MySQL develops almost all code in-house. Valimaki (2003) finds that the company even asks its own employees to rewrite the contributed code from external developers such as bug fixes to clean copyright ownership. In this sense, MySQL can hardly enjoy the contribution from open source community like Red Hat even though it releases the source code.

3 Licensing Schemes as Differentiation Tools

Varian (1989) argues that the producer should provide different versions of a software product which sell at different prices to maximize the profits. It is quite popular for proprietary software companies to differentiate their software products to maximize the profits. For example, Windows Vista have six editions targeted to different consumers[7]. At the same time, software companies can also apply licensing schemes to realize product differentiation besides quality. The potential users in the market normally have heterogeneous preference on the rights granted by copyright holders. Average users just need the necessary right to run the programme to realize specific functions. But developers with high computing skills need the rights to modify software besides basic using rights. At last there are also some commercial companies who want to get redistribution rights to integrate the software with their own programme. When software producers choose proprietary licensing schemes, he normally will only provide two types of licenses to average users and integrators respectively. The reason that why developers are not served by proprietary software companies is that implementing

[7] http://www.microsoft.com/presspass/press/2006/feb06/02-26WinVistaProductsPR.mspx

modification is extremely difficult without access source code, which is strictly protected under proprietary licenses.

On the other hand, software companies can enjoy saving on development cost if they choose open source licensing schemes. But correspondingly they will give away most rights granted by copyright law, such as using right, modification right and redistribution right. In this case, software companies will be incapable of using licensing scheme to carry out differentiation at all. However, open source licensing schemes may dramatically increase software's adoption base. At the same time, when average consumers use software as a kind of consumption products, they may also care about the necessary support and service besides the code quality. When the consumers are lack of computer knowledge and programming skills, or the software's maintenance and upgrading are so complicated, necessary technical support and service are important and even dispensable for the software users to maximize the potential utility from the software product. According to the definition of open source software[8], the distribution of open source software only guarantees the free availability of source code as well as complied form of the programme. The developers and distributors have no obligation to provide technical support or other necessary services. Lerner & Tirole (2002) argues that lack of necessary documentation, technical support and friendly interface is main barriers for beginner users's adoption of open source software. All of these issues can be easily solved when customer-oriented firms to provide such service. Users will be willing to pay for the necessary support or complementary enhancements. When there exists positive network externality, profits from providing value added service will increased with expanded adoption base.

In addition to pure proprietary and pure open source licensing schemes, software companies can choose the mixed licensing schemes, that is, licensing the software under both proprietary and open source licenses. Actually Bonaccorsi, Giannangeli & Rossi (2006) shows that the large majority of firms choose hybrid business model by mixing products, licenses and source of revenues by analyzing the data from a survey on 146 Italian software firms. They think this is an adaption of open source to the dominant proprietary standard. The advantage of mixed licensing schemes is that the licensor can obtain larger market share with open source software than pure proprietary schemes without sacrificing product

[8] See the defition from Open Source Initiative in http://www.opensource.org/docs/osd

differentiation capability as pure open source licensing schemes. However, mixed licensing schemes may damage the cost saving effect of releasing source code. As mentioned above in MySQL case, mixed licensing schemes require that firms have full control over copyright of software. So the firms who adopt mixed licensing schemes will either clean copyright ownership by rewriting just like MySQL or they need to add terms into the open source license to ask the contributors to transfer the copyrights to the firms, which will probably compromise the relation between communities and firms. Either way will make the firms lose cost advantage which they can obtain under pure open source licensing schemes. At last, reciprocity terms in open source licenses are worth further discussion here. The reciprocity terms in open source licenses provide a protection against commercial hijacking. They are used to prevent that the original code is modified and redistributed under a proprietary license. Otherwise, benefits generated by voluntary programmers can be easily privatized and internalized. The reciprocity terms are only effective in the case where firms want to integrate code of open source software into their commercial software to make profits. The characteristics of the reciprocity terms make them important at product differentiation when mixed licensing schemes are adopted. If a commercial company tries to integrate code of open source software licensed with the reciprocity terms, he has to disclose the whole source code of the big project after integration. This virus nature will induce the integrators to buy a proprietary license instead of use the free open source version in the case of mixed licensing schemes.

So the choice of different licensing scheme depends on the profits from product differentiation and development cost under each licensing scheme. In the next section, a theoretical model will be presented to show the details.

4 Model

Suppose that a commercial software firm plans to develop and distribute a new software product targeted to a group of heterogeneous consumers. These consumers are interested not only the code quality but also licensing schemes.

The firm can follow three alternative licensing schemes to develop and distribute the software product: pure proprietary, pure open source and mixed. Once pure proprietary licensing scheme is chosen, the firm can only develop the software project on its own and distribute it under the proprietary license. If pure open source strategy is adopted, it releases the code to the open source community and enjoy the free contribution to save R&D cost during the development stage then must distribute the completed software product under the open source license. At last, the firm can also choose mixed licensing scheme, that is, he develops the project alone but distribute the software under dual licenses, both open source and proprietary[9].

The firm takes its decisions sequentially and the timing of game is as follow:

1. Development Stage: the firm needs to decide whether to develop the software project all by himself or release the source code to the open source community. If he chooses the latter, he must also choose the appropriate open source license. For the sake of simplicity, I assume the firm only needs to choose the reciprocity degree t, where $t \in [0, 1]$ [10]. Here I also assume that the code quality of the software is the same for both methods. The development method is irrelevant to the quality of final software product, it only changes the cost. The developing cost of software is $C \geq 0$ if the firm develops it alone and zero if he chooses to release the source code to the open source community. Here I assume the firm can perfectly disperse the development cost to the outside researchers. To simplify the analysis, I also normalize the cost of providing support and service to zero.

2. Distribution Stage: the firm now chooses the license for the software product if it is developed on his own in the development stage. At the same time, he may also need to choose the price if the software is distributed under the proprietary license.

Developers from open source community, with a population of 1, will not adopt the software unless the firm releases the source code. Different timing of releasing the source code will have different effect on the cost-saving and copyright

[9] Here, I assume that single development is the precondition of full control over the copyright of software products. Once the project is developed with open source method, the firm will lose its full copyright over the project because the each contributor own the copyright on his contributed code. In other words, releasing source code at the development stage will force the firm to distribute the software product only under the open source license.

[10] Though there are so many different open source licenses in the market, most of them have similar terms of free distribution of the programme and its source code. The diversification are normally originated from the terms on the redistribution of derivative works. The higher reciprocity degree the license has, the more limitation it imposes on the distribution of derivative works.

control. Early release at the development stage will make the firm enjoy the free contribution from the developers but lose full control of copyright. Late release at the distribution stage can give the firm more flexibility of licensing by sacrificing the cost-saving benefit from open source community.

Consumers observe the firm's licensing choice at the distribution stage and make their own adoption decisions[11]. There are two kinds of consumers in the market, the average users and the integrators. Each of them has a population of 1.

Assumption 1: The usage of the software has positive and asymmetric network effect, which is stronger for the integrators than the average users.

Assumption 2: The firm bundles the support with the programme when the software product is distributed under the proprietary license. However, support can be purchased separately under the open source distribution.

For the software product with certain code quality and distributed under the proprietary license, the net utility of an average user is

$$U_s^P = \begin{cases} (n-1)(1+\gamma) - p_{cs} \text{ if he purchases the proprietary software} \\ 0 \text{ if he does not adopt the software} \end{cases}$$

If it is distributed as open source software, then the utility is

$$U_s^O = \begin{cases} (n-1) \text{ if he adopts the open source software without support} \\ (n-1)(1+\gamma) - p_s \text{ if he adopt the open source software with support} \\ 0 \text{ if he does not adopt the software} \end{cases}$$

The utility of adopting a software product with certain quality will also depend on the user base, n. The average users can get extra utility from the support and service. Here I use $\gamma \geq 0$ to measure their willing-to-pay for the support. When γ is close to zero, the average users are more support-independent and when γ is increasing, they are more support-oriented. p_{cs} and p_s here are the prices charged by the firm for the software bundle and support respectively.

For a integrator, his utility under the proprietary license is equal to

[11] I rule out the possibility of early adoption by the consumers. It is always optimal for both the average users and the integrators to wait for well-developed mature version of the software product to avoid the high uncertainty and risk.

$$U_I = \begin{cases} n - p_I \text{ if he buys the software under the proprietary license} \\ n(1-\beta t) \text{ if he adopts the software under the open source license} \\ 0 \text{ if he does not adopt the software} \end{cases}$$

Where $\beta \geq 0$ is a measurement of how sensitive the firm are to the reciprocity degree t. When β is larger, the disutility from the reciprocal terms are bigger even for the same reciprocity degree. In this case, the embedded code is playing a core role in the new software product or the most codes to be integrated with are under the proprietary license. If a integrator embeds the original code with other open source code or the embedded code is less relevant to the system, he is not significantly hurt by the reciprocal terms in the license, which is characterized by a smaller β. The integrators are reciprocity-sensitive when β is large and reciprocity-insensitive when β is approaching zero.

4.1 The Optimal Licensing Scheme

When the firm chooses the pure proprietary licensing scheme, he will charge

$$\begin{cases} p_{cs} = 1 + \gamma \\ p_I = 2 \end{cases}$$

and both the average users and the integrators will adopt the software product. So the profit is

$$\pi^P = 3 + \gamma - C$$

When the firm chooses the pure open source licensing scheme, he must distribute the software product only at the open source license, which means he can not earn any profits from the integrators. However, he can still manipulate the adoption decision of the integrators by choosing the reciprocity degree t. It is always optimal for him to choose $t \in \left[0, \min\left\{\frac{1}{\beta}, 1\right\}\right]$ to make sure all the integrators will adopt the software product. Therefore, he can charge higher price for the support from the average users because of the network effect. At the same time, the developers from open source community will also adopt the software and provide free contributions. Then the firm will set

$$p_s = 2\gamma$$

and the total user base now is 3. The profit is

$$\pi^O = 2\gamma$$

When the firm chooses the mixed licensing scheme, he distributes the software at two different licenses: both proprietary and open source. To achieve this licensing flexibility, he sacrifices the development benefit from the open source community. However, late release of source code can still make the outside developers part of user base. Now the firm will set

$$p_s = 2\gamma \text{ and } p_I = \begin{cases} 3 \text{ if } \beta > 1 \\ 3\beta \text{ if } 0 \leq \beta \leq 1 \end{cases}$$

The reciprocity degree of the open source license will be

$$t \in \left[\min\left\{\frac{1}{\beta}, 1\right\}, 1\right]$$

The profit under the mixed licensing scheme is

$$\pi^M = \begin{cases} 2\gamma + 3 - C \text{ if } \beta > 1 \\ 2\gamma + 3\beta \ - C \text{ if } 0 \leq \beta \leq 1 \end{cases}$$

According to the above analysis of each licensing scheme, the following proposition can be reached:

Proposition 1 *a). If the firm chooses the pure open source licensing scheme, the license is more permissive when the integrators are reciprocity-sensitive; however, when the integrators are reciprocity-insensitive, the reciprocity degree of the open source license is diversified.*

b). If the firm adopts the mixed strategy, the license is more restrictive when the integrators are reciprocity-insensitive; however, when the integrators are reciprocity-sensitive, the reciprocity degree of the open source license is diversified.

Proof. When the pure open source licensing scheme is chosen, the firm sets

$$t \in \left[0, \min\left\{\frac{1}{\beta}, 1\right\}\right]$$

to maximize the profits. When $\beta \to \infty$, $\min\left\{\frac{1}{\beta}, 1\right\} \to 0$ and $t \to 0$; when $\beta \leq 1$, $\min\left\{\frac{1}{\beta}, 1\right\} = 1$ and $t \in [0, 1]$.

Similarly

$$t \in \left[\min\left\{\frac{1}{\beta}, 1\right\}, 1\right]$$

if the mixed licensing scheme is adopted. When $\beta \to \infty$, $\min\left\{\frac{1}{\beta}, 1\right\} \to 0$ and $t \in [0, 1]$; when $\beta \leq 1$, $\min\left\{\frac{1}{\beta}, 1\right\} = 1$ and $t = 1$. ■

When the pure open source licensing scheme is adopted, the firm can only earn profits by selling support and service to the average users. Because of the network effect, the user's utility is enhanced with an increase in the total population of software users. Then the average users are willing to pay more for the same software product with same support and service. To induce the adoption by the integrators, the firm must choose a reciprocity degree t, which makes the integrators have positive utility by adopting the open source software. If the integrators are less sensitive to the reciprocal terms, then the choice of reciprocity degree will be quite trivial. Otherwise, the firm has to make the open source license as permissive as possible. When the firm chooses the mixed licensing scheme, the integrators are provided by two different licenses. To motivate them to purchase the proprietary license, the firm will try to make the open source version less attractive by adding reciprocal terms. When the integrators are reciprocity-insensitive, the reciprocity degree will be higher to guarantee the effect. But if the integrators are reciprocity-sensitive, the manipulating space is large, which leads to the possibility of diversification of licensing.

By comparing the profits from each licensing scheme, I can obtain the following proposition.

Proposition 2 *As long as $C \geq 3$, pure open source licensing scheme is always the optimal one.*

Proof. When $C \geq 3$, it is easy to get that

$$\pi^M \leq 2\gamma + 3 - C \leq 2\gamma = \pi^O$$

and

$$\begin{aligned} \pi^O - \pi^P &= 2\gamma - (3 + \gamma - C) \\ &= \gamma + (C - 3) \geq 0 \end{aligned}$$

■

So when the development cost for the software project is high, pure open source licensing scheme is always optimal because cost-saving is the dominant effect. This also explains why so many start-up firms are pioneering in the open source business. For example, the development of operating system software is quite complicated and costly. Wheeler (2001) estimates that without the contributes from the open source community, the Red Hat 7.1 operating system would have cost about one billion dollars, which is far more affordable for small or medium companies.

Now let's consider the case where the development cost is not so high, that is, $0 \leq C \leq 3$.

Proposition 3 *If* $0 \leq C \leq 3$ *and* $\beta \geq 1$*, the mixed licensing scheme will be dominant.*

Proof. If $0 \leq C \leq 3$, then

$$\pi^O = 2\gamma \leq 2\gamma + 3 - C = \pi^M$$

and

$$\pi^M - \pi^P = \gamma \geq 0$$

■

So the firm always prefers the mixed licensing scheme as long as the integrators are reciprocity-sensitive. When the $\beta \geq 1$, the firm can choose an appropriate reciprocity degree t such that the integrators only obtain negative utility from adoption of open source version. They must pay full price to get the proprietary version. In this situation, production differentiation dominates cost saving effect.

The most interesting case is for $0 \leq C \leq 3$ and $\beta < 1$. In this situation, the benefit from releasing the source code on cost-saving is limited compared with the monopoly profits from keeping the source code closed. At the same time, profits from selling the proprietary license to the integrators are compromised because they are less reluctant to the reciprocal terms in the open source license. Now the decision of the optimal licensing scheme is determined by the β and γ together for any given C.

Proposition 4 *For any give development cost $C \in [0,3]$ and $\beta < 1$, the firm will always choose not to release the source code as long as $\gamma \leq \min\{3-3\beta, 3-C\}$; otherwise, the firm is better off to release if $\gamma \geq \min\{3-3\beta, 3-C\}$. Specifically, he will choose the mixed licensing scheme when $\frac{C}{3} \leq \beta < 1$ and the pure open source licensing scheme when $0 < \beta < \frac{C}{3}$.*

Proof. By comparing the profits from each licensing scheme, it is easily to get that for the development cost $C \in [0,3]$ and $\beta < 1$,

$$\pi^O \geq \pi^M$$

when $0 < \beta < \frac{C}{3}$. Comparing π^O with π^P leads to that

$$\begin{cases} \pi^O \geq \pi^P \text{ when } \gamma \geq 3-C \\ \pi^O \leq \pi^P \text{ when } \gamma < 3-C \end{cases}$$

Similarly, for $\frac{C}{3} \leq \beta < 1$,

$$\begin{cases} \pi^M \geq \pi^P \text{ when } \gamma \geq 3-3\beta \\ \pi^M \leq \pi^P \text{ when } \gamma < 3-3\beta \end{cases}$$

■

Actually this result is quite interesting, it shows that when the development cost is moderate and the integrators are not extremely reciprocity-sensitive, the choice of whether to release the source code is determined by the value of γ and the choice of timing of release is determined by β. When γ is large, the average users are more support-oriented so the support is more valuable to them. Therefore the firm can still generate enough profits by giving away the programme for free. Combined with the benefits from saving on development cost and network effect, the firm are always better off by releasing the source code. At the same time, the timing of the release, in other words, choice between the pure open source and mixed licensing scheme, is determined by β. Larger β means the integrators are more reciprocity-sensitive, and the firm can earn more profits from differentiate the licenses at the cost of free contribution from the open source community because of the late release, compared with early release will help the firm save research cost but deprive its ability of product differentiation. On the contrary, the firm will always choose not to release the source code and distribute the software product under the proprietary license as long as γ is small, that is, the consumers are support-independent. The relation between choice of licensing scheme and the characteristics of consumers when development cost is not high ($0 \leq C \leq 3$) can be summarized as follow:

Table I: Choice of Licensing Schemes

	$0 < \beta < \frac{C}{3}$	$\frac{C}{3} \leq \beta < 1$	$\beta \geq 1$
$\gamma \leq \min\{3 - 3\beta, 3 - C\}$	Pure Proprietary	Pure Proprietary	Mixed
$\gamma > \min\{3 - 3\beta, 3 - C\}$	Pure Open Source	Mixed	Mixed

4.2 Conclusion

An increasingly larger populations of software companies are getting involved into the open source business. And coexistence of open source and proprietary software seems to be normal in the software industry. Cases of success keep on rising from both sides. So it is really interesting and important to understand which is the optimal licensing scheme for a for-profits firm: pure proprietary, pure open source or mixed?

This paper proposes a theoretical model to study the choice of optimal licensing scheme. In the model, the software monopolist need to consider both profits generated by product differentiation of licensing scheme and its influence on development cost. The results suggest that as long as the usage of software shows network externality, high development cost will induce the firm to choose the pure open source license. When the development cost is moderate, the pure proprietary license is more preferred when the average users are support-independent, otherwise the firm is willing to release the source code when they are support-oriented. The more sensitive are the integrators to the reciprocal terms in the open source license, the more likely the pure open source licensing scheme is dominated by the mixed strategy.

At the same time, this paper also provides a reasonable explanation for the diversification of open source licensing. Generally speaking, firms who choose the pure open source licensing scheme are inclined to adopt permissive license while those follow the mixed licensing scheme are more likely to adopt the restrictive license. However, the manipulation space of reciprocity degree sometimes may be quite large, which result in the diversification of open source licensing.

References

Bitzer, J. (2004), ‘Commercial versus open source software: The role of product heterogeneity in competition’, *Economic Systems* **28(4)**, 369–381.

Bonaccorsi, A., Giannangeli, S. & Rossi, C. (2006), ‘Entry strategies under competiting standard: Hybrid business models in the open source software industry’, *Management Science* **52(7)**, 1085–1098.

Bonaccorsi, A. & Rossi, C. (2003), ‘Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business’, SSRN working paper.

Casadesus-Masanell, R. & Ghemawat, P. (2006), ‘Dynamic mixed duopoly: A model motivated by linux vs. windows’, *Management Science* **52(7)**, 1072–1084.

Dahlander, L. & Magnusson, M. G. (2005), ‘Relationships between open source software companies and communities: Observations from nordic firms’, *Research Policy* **34(4)**, 481–493.

Dahlander, L. & Magnusson, M. G. (2006), Business models and community relationships of open source software firms, *in* J. Bitzer & P. J. H. Schroder, eds, ‘The Economics of Open Source Software Development’, Elsevier, pp. 111–130.

Economidies, N. & Katsamakas, E. (2006), ‘Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry’, *Management Science* **52(7)**, 1057–1071.

Feller, J. & Fitzgerald, B. (2002), *Understanding Open Source Software Development*, Addison Wesley, Boston, MA.

Hars, A. & Ou, S. (2002), ‘Working for free? motivations for participating in open-source projects’, *International Journal of Electronic Commerce* **6**, 25–39.

Lerner, J. & Tirole, J. (2002), ‘Some simple economics of open source’, *Journal of Industrial Economics* **50**, 197–234.

Lerner, J. & Tirole, J. (2005), 'The scope of open source licensing', *The Journal of Law, Economics and Organization* **21**, 20–56.

Narduzzo, A. & Rossi, C. (2003), 'Modularity in action: GNU/Linux and Free/Open source software development model unleashed', ROCK Working Papers 020.

Raymond, E. S. (2000), 'The cathedral and the bazaar', http://www.catb.org/esr/writings/cathedralbazaar/cathedral-bazaar, accessed on July 19th, 2005.

Rossi, C. & Bonaccorsi, A. (2006), Intrinsic motivations and profit-oriented firms in open source software. do firms practise what they preach?, *in* J. Bitzer & P. J. H. Schroder, eds, 'The Economics of Open Source Software Development', Elsevier, pp. 83–110.

Valimaki, M. (2003), 'Dual licensing in open source software industry', Systemes d'Information et Management 1/2003.

Valimaki, M. (2005), *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property in the Software Industry*, Turre Publishing.

Varian, H. R. (1989), Price discrimination, *in* R. Schmalensee & R. Willig, eds, 'Handbook of Industrial Organization', North-Holland Press, Amersterdam.

Von Hippel, E. (2003), 'Open source projects as horizontal innovation networks - by and for users', MIT Sloan School of Management Working Paper 4366-02.

Von Krogh, G. & Von Hippel, E. (2006), 'The promise of research on open source software', *Management Science* **52(7)**, 975–983.

Wheeler, D. (2001), 'More than a gigabuck: Estimating GNU/Linux's size', http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html.

Wichmann, T. (2002), 'Firms' open source activities: Motivations and policy implications', Free/Libre and Open Source Software: Survey and Study, FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

SCHOOL OF ECONOMICS AND MANAGEMENT
UNIVERSITY OF AARHUS - UNIVERSITETSPARKEN - BUILDING 1322
DK-8000 AARHUS C – TEL. +45 8942 1111 - www.econ.au.dk

PhD Theses:

1999-4	Philipp J.H. Schröder, Aspects of Transition in Central and Eastern Europe.
1999-5	Robert Rene Dogonowski, Aspects of Classical and Contemporary European Fiscal Policy Issues.
1999-6	Peter Raahauge, Dynamic Programming in Computational Economics.
1999-7	Torben Dall Schmidt, Social Insurance, Incentives and Economic Integration.
1999	Jørgen Vig Pedersen, An Asset-Based Explanation of Strategic Advantage.
1999	Bjarke Jensen, Five Essays on Contingent Claim Valuation.
1999	Ken Lamdahl Bechmann, Five Essays on Convertible Bonds and Capital Structure Theory.
1999	Birgitte Holt Andersen, Structural Analysis of the Earth Observation Industry.
2000-1	Jakob Roland Munch, Economic Integration and Industrial Location in Unionized Countries.
2000-2	Christian Møller Dahl, Essays on Nonlinear Econometric Time Series Modelling.
2000-3	Mette C. Deding, Aspects of Income Distributions in a Labour Market Perspective.
2000-4	Michael Jansson, Testing the Null Hypothesis of Cointegration.
2000-5	Svend Jespersen, Aspects of Economic Growth and the Distribution of Wealth.
2001-1	Michael Svarer, Application of Search Models.
2001-2	Morten Berg Jensen, Financial Models for Stocks, Interest Rates, and Options: Theory and Estimation.
2001-3	Niels C. Beier, Propagation of Nominal Shocks in Open Economies.
2001-4	Mette Verner, Causes and Consequences of Interrruptions in the Labour Market.
2001-5	Tobias Nybo Rasmussen, Dynamic Computable General Equilibrium Models: Essays on Environmental Regulation and Economic Growth.

2001-6	Søren Vester Sørensen, Three Essays on the Propagation of Monetary Shocks in Open Economies.
2001-7	Rasmus Højbjerg Jacobsen, Essays on Endogenous Policies under Labor Union Influence and their Implications.
2001-8	Peter Ejler Storgaard, Price Rigidity in Closed and Open Economies: Causes and Effects.
2001	Charlotte Strunk-Hansen, Studies in Financial Econometrics.
2002-1	Mette Rose Skaksen, Multinational Enterprises: Interactions with the Labor Market.
2002-2	Nikolaj Malchow-Møller, Dynamic Behaviour and Agricultural Households in Nicaragua.
2002-3	Boriss Siliverstovs, Multicointegration, Nonlinearity, and Forecasting.
2002-4	Søren Tang Sørensen, Aspects of Sequential Auctions and Industrial Agglomeration.
2002-5	Peter Myhre Lildholdt, Essays on Seasonality, Long Memory, and Volatility.
2002-6	Sean Hove, Three Essays on Mobility and Income Distribution Dynamics.
2002	Hanne Kargaard Thomsen, The Learning organization from a management point of view - Theoretical perspectives and empirical findings in four Danish service organizations.
2002	Johannes Liebach Lüneborg, Technology Acquisition, Structure, and Performance in The Nordic Banking Industry.
2003-1	Carter Bloch, Aspects of Economic Policy in Emerging Markets.
2003-2	Morten Ørregaard Nielsen, Multivariate Fractional Integration and Cointegration.
2003	Michael Knie-Andersen, Customer Relationship Management in the Financial Sector.
2004-1	Lars Stentoft, Least Squares Monte-Carlo and GARCH Methods for American Options.
2004-2	Brian Krogh Graversen, Employment Effects of Active Labour Market Programmes: Do the Programmes Help Welfare Benefit Recipients to Find Jobs?
2004-3	Dmitri Koulikov, Long Memory Models for Volatility and High Frequency Financial Data Econometrics.
2004-4	René Kirkegaard, Essays on Auction Theory.

2004-5 Christian Kjær, Essays on Bargaining and the Formation of Coalitions.

2005-1 Julia Chiriaeva, Credibility of Fixed Exchange Rate Arrangements.

2005-2 Morten Spange, Fiscal Stabilization Policies and Labour Market Rigidities.

2005-3 Bjarne Brendstrup, Essays on the Empirical Analysis of Auctions.

2005-4 Lars Skipper, Essays on Estimation of Causal Relationships in the Danish Labour Market.

2005-5 Ott Toomet, Marginalisation and Discouragement: Regional Aspects and the Impact of Benefits.

2005-6 Marianne Simonsen, Essays on Motherhood and Female Labour Supply.

2005 Hesham Morten Gabr, Strategic Groups: The Ghosts of Yesterday when it comes to Understanding Firm Performance within Industries?

2005 Malene Shin-Jensen, Essays on Term Structure Models, Interest Rate Derivatives and Credit Risk.

2006-1 Peter Sandholt Jensen, Essays on Growth Empirics and Economic Development.

2006-2 Allan Sørensen, Economic Integration, Ageing and Labour Market Outcomes

2006-3 Philipp Festerling, Essays on Competition Policy

2006-4 Carina Sponholtz, Essays on Empirical Corporate Finance

2006-5 Claus Thrane-Jensen, Capital Forms and the Entrepreneur – A contingency approach on new venture creation

2006-6 Thomas Busch, Econometric Modeling of Volatility and Price Behavior in Asset and Derivative Markets

2007-1 Jesper Bagger, Essays on Earnings Dynamics and Job Mobility

2007-2 Niels Stender, Essays on Marketing Engineering

2007-3 Mads Peter Pilkjær Harmsen, Three Essays in Behavioral and Experimental Economics

2007-4 Juanna Schrøter Joensen, Determinants and Consequences of Human Capital Investments

2007-5 Peter Tind Larsen, Essays on Capital Structure and Credit Risk

2008-1 Toke Lilhauge Hjortshøj, Essays on Empirical Corporate Finance – Managerial Incentives, Information Disclosure, and Bond Covenants

2008-2 Jie Zhu, Essays on Econometric Analysis of Price and Volatility Behavior in Asset Markets

2008-3 David Glavind Skovmand, Libor Market Models - Theory and Applications

2008-4 Martin Seneca, Aspects of Household Heterogeneity in New Keynesian Economics

2008-5 Agne Lauzadyte, Active Labour Market Policies and Labour Market Transitions in Denmark: an Analysis of Event History Data

2009-1 Christian Dahl Winther, Strategic timing of product introduction under heterogeneous demand

2009-2 Martin Møller Andreasen, DSGE Models and Term Structure Models with Macroeconomic Variables

2009-3 Frank Steen Nielsen, On the estimation of fractionally integrated processes

2009-4 Maria Knoth Humlum, Essays on Human Capital Accumulation and Educational Choices

2009-5 Yu Wang, Economic Analysis of Open Source Software