View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Research Papers in Economics

Open Source vs. Proprietary Software: Competition and Compatibility

Zhaoli Meng
Department of Information Systems, National University of Singapore
3 Science Dr. 2, Singapore, 117543

Sang-Yong Tom Lee[1]
College of Information and Communications, Hanyang University
17 Hangdang-dong, Seongdong-gu, Seoul 133-791, Korea

Abstract

We use a Hotelling linear city model to study competition between open source and proprietary software, where only the producer of the proprietary software aims at maximizing the profit. The producer of the proprietary software must decide on compatibility. Different compatibility strategies will lead to different network externality, and thus result in different profit for the producer of the proprietary software. We found that the proprietary producer's choice of compatibility strategy depends on the market coverage conditions. When the market is fully covered, one-way compatibility is the best strategy for the proprietary software. When the market is partly covered, two-way compatibility is the best strategy. Such results are not affected by software quality. Furthermore, when the provider of the open source software pursues the maximum market share rather than reacts passively, two-way compatibility would be the best choice for both the open source and the proprietary software. Moreover, the proprietary software producer does not favor its proprietary rival changing to open source software. Such a change may lower the social welfare.

Key Words: Open Source Software, Proprietary Software, Compatibility, Competition

JEL Codes: L1, L13, L86

[1] Corresponding author. Email: tomlee@hanyang.ac.kr. The authors gratefully acknowledge Ivan Png for his helpful comments and suggestions.

1. INTRODUCTION

Recent years have seen the striking success of open source software, which allows software developers to use shared source codes, identify and correct errors, and redistribute the source codes (OSI 2001, O'Reilly 1999). One of the most famous and successful open source software projects is Linux, which commands a third market share within the server operating system market, and whose share is expected to grow to 41 percent by 2005 (International Data Corporation IDC 2002). Another well known example is Apache, which supports 67 percent of web sites on the Internet (Netcraft Web Server Survey 2004). Other successful open source software products have had significant market shares in their product categories. For instance, Sendmail, an open source email transfer program, carries an estimated 80% of the entire world's e-mail traffic (Weber 2004).

Open source software threatens proprietary software producers.[2] Amazon reported that adopting open source software has reduced 25 percent of its technology expenses. In the city of Europe, Munich is switching from Windows to Linux, and from Internet Explorer (IE) to Mozilla browser (CNet News 2004a).

To survive and win the maximum profit in the battle with the open source software, proprietary software producers may adopt one of four different compatibility strategies, i.e., incompatibility, two-way compatibility, inward compatibility, and outward compatibility.[3] These four compatibility strategies can be understood through the following examples:

- Windows, a proprietary software product, is incompatible with Linux, an open source software product.
- In the case of web browsers, Internet Explorer, a proprietary software product, is two-way compatible with Mozilla, an open source product. Files created for IE users can be used by Mozilla users and vice versa.

[2] We use the term 'proprietary software' as non-free software (Working Group on Libre Software 2000)

[3] Katz and Shapiro (1998) defined compatibility as follows: "When two programs can communicate with one another and/or be used with the same complementary system components, they are said to be compatible".

- In the web server market, Microsoft IIS (a proprietary web server) is inward compatible with Apache (an open source web server): Microsoft IIS can support both PHP and ASP, server side programming languages. Thus, programs designed for Apache using PHP language can be executed in IIS. However, since ASP belongs to Microsoft and can not execute on Apache, the programs designed for IIS using ASP are not usable in Apache.

- Finally, outward compatibility means that the files or programs designed for the proprietary software can be used by the open source software while files or programs designed for the open source software are not usable by the proprietary software. Realistically, outward compatibility is seldom observed.

These different compatibility strategies present a series of research questions: why would a proprietary software producer choose different strategies of compatibility when facing competition from open source software? How would the choices of compatibility affect the profitability of a proprietary software producer? Furthermore, what are the welfare implications?

Another series of questions relates to the impact of open source software. If the open source software pursues maximum market share rather than reacts passively, should the proprietary software producer change it compatibility strategy? How would the proprietary producer's profit, price and market share be affected if its rival switches from proprietary software to open source software? Does such a switch benefit society or not?

This paper addresses these research questions using Hotelling's model of competition between the open source and proprietary software. In contrast with the conventional Hotelling model, only one party – the proprietary software producer – aims at maximizing profit, and the open source software is passive.

The different compatibility choices of the proprietary software result in different network externalities for the open source and proprietary software. Thus, the price and profit of the proprietary software varies correspondently. We compare the maximum profit of the proprietary software under different compatibility strategies, and propose the best compatibility strategy for the propriety software producer.[4]

The main findings in this part are: When the market is fully covered, i.e., when all the consumers purchase one of the two products, inward compatibility is the best strategy. However, when the market is partly covered, two-way compatibility is the best strategy. Furthermore, the welfare analysis provides some implications on how the welfare would be affected by different parameters, such as the network externality intensity and software quality.

Next, we relax the conditions of the basic model to investigate competition between open source and proprietary software under different scenarios. Firstly, we suppose that the open source software aims at maximizing the marker share rather than reacts passively. In such a case, two-way compatibility is a Nash Equilibrium in both fully covered market and partly covered market.

Secondly, we examine the impact of open source software on proprietary software and on society. It is found that a proprietary software producer does not favor its proprietary rival changing to open source software because such change will lower its market share, price and profit. Furthermore, contrary to the general belief that the change from proprietary software to open source software will benefit society, we find that under certain conditions, such a change will decrease social welfare.

The rest of this paper is organized as follows. Section 2 reviews the literature on open source software. Section 3 introduces the basic Hotelling model of asymmetric competition between open source and proprietary software. In sections 4, we compare the market results under different strategy choices when the base-level qualities of the open source and proprietary software are same. Section 5 presents market results when open source and proprietary software have different base-level qualities. Section 6 proposes the best compatibility strategy under different market coverage conditions. Section 7 investigates the implication for welfare. Section 8 extends the basic model to investigate the competition between open source and

[4] In the following context, by default, the best compatibility strategy means the best compatibility strategy for a proprietary software producer.

proprietary software under different scenarios. Section 9 discusses the results and suggests possible directions for future work.

2. Literature Review

The most widely investigated research question in past literature on open source software is to identify the economic and non-economic motivations for individual developers to contribute to open source software (Lerner and Tirole 2002, Lakhani and Wolf 2003, Hann et al. 2002). Currently, researchers classify the possible reasons into intrinsic motivation, such as intellectual stimulation (Lakhani and Wolf 2003), and extrinsic motivation, including career concerns (Lerner and Tirole 2002) and peer recognition (Raymond 1999, Vostroknutov 2002).

Another theme of prior research focuses on the quality of open source software and the competition between open source and proprietary software. An important conclusion is that open source software is not necessarily inferior in quality to proprietary software (Mishra et al. 2002, Dalle and Jullien 2002, Kuan 2001, Johnson 2001, Bessen 2002). This conclusion is derived from models of different aspects: Mishra et al. (2002) compared the quality of software under open source and closed environments; Kuan (2001) demonstrated that open source software has a higher rate of quality improvement than proprietary software; Johnson (2001) modeled open source software as the private provision of public goods; Dalle and Jullien (2002) presented organizational structure and compatibility as key factors to the quality of open source software.

Within the research theme outlined above, one strand has been to consider the competition between open source and proprietary software. Casadesus-Masanell and Ghemawat (2003) modeled the competition between Windows and Linux as a dynamic “mixed duopoly”, where a not-for-profit competitor interacts with a for-profit competitor. What the study named mixed duopoly differs from the well-investigated mixed oligopoly competition, where one party pursues profit maximization while the other (most probably a public producer) aims at welfare maximization (Cremer et al. 1989, 1991, DeFraja and Delbono 1989, Fershtman 1990, Fjell and Pal 1996, White 1996). Casadesus-Masanell and Ghemawat (2003) showed that, as long as Windows' pricing decision was not myopic, the result of the competition would be either the coexistence of the two products or Linux being driven out of the market.

Our paper takes a similar approach but differs from Casadesus-Masanell and Ghemawat (2003) in two aspects. The main difference is in the research questions. Our work focuses on strategic choices for compatibility – a topic seldom investigated in previous studies on open source software. We seek to find the best compatibility strategy rather than predict the results of competition. Furthermore, we model the consumers' heterogeneous preferences for products, which depend on two factors: the learning cost of adopting a software product varies for different consumers, and the difference in their past experiences, i.e., the extent of 'lock-in' is different among consumers and their switching costs are different. Due to these two factors, consumers have different tastes for the products. The consumers who have lower taste for the open source software would rather choose the proprietary software even though the open source software is free of charge. The difference in the consumers taste allows rich intuitive interpretations for real-world software competition.

3. Basic Model

Consider a software market where two software products are located at the ends of a unit line, i.e., the open source software (O) is located at $x=0$ and the proprietary software (P) is located at $x=1$ (see Figure 1). Consumers are uniformly distributed along the unit interval and they have unit demand for the software. Consumers differ in their taste for the products. Specifically, for a consumer located at $x \in [0,1]$, she incurs utility cost tx if she uses the open source software because of the difference between her ideal preference and the product specifications. Similarly, she incurs utility cost $t(1\text{-}x)$ if she chooses the proprietary software, where t measures the consumers' taste difference. We assume that the marginal costs of both the open source and proprietary software products are zero. Following Katz and Shapiro

(1985), the network externalities are a linear function of the number of users who adopt the same or compatible software products.

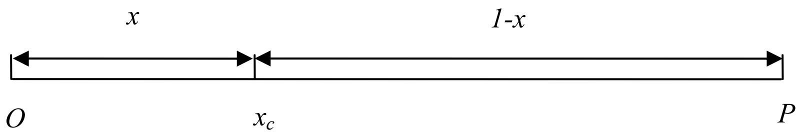

Figure 1: The Basic Hotelling Model

In the basic model, we assume that the two products have the same inherent quality *s* and are incompatible, and the market is fully covered, i.e., all the consumers choose to use one of the two software products. This is always true when the benefit of the product is sufficiently large. If a consumer located at *x* adopts the open source software, her net utility U_o would be $U_o = s + kq_o - tx + \gamma q_o$, where q_o is the number of open source users and k is the degree of contribution of each consumer to the quality of the open source software. The parameter γ refers to the network externality that a software user receives from other users of same or compatible software. We assume that the open source software product is freely available, and there is no price component in the net utility. Similarly, if the consumer located at *x* adopts the proprietary software, her net utility U_p is $U_p = s - t(1-x) + \gamma q_p - p$, where q_p denotes the number of proprietary software users and *p* is the price of the proprietary software.

Suppose the consumer at $x_c \in [0,1]$ is indifferent between the open source and proprietary software products, then from $U_o = U_p$, we have:

$$x_c = \frac{p+t-\gamma}{2t-2\gamma-k}, q_o = \frac{p+t-\gamma}{2t-2\gamma-k}, q_p = 1 - \frac{p+t-\gamma}{2t-2\gamma-k}.$$

From $0 < x_c < 1$, and applying the second-order condition for profit maximization, we have:

$p+t-\gamma>0$, and $2t-2\gamma-k>0$.

The profit for the proprietary software producer is:

$$\pi(p)=pq_p=p(1-x_c)=p[1-\frac{p+t-\gamma}{2t-2\gamma-k}].$$

By solving the profit maximization problem with respect to *p*, we get:

$$p^*=\frac{t-\gamma-k}{2},\quad \pi^*=\frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]}$$

, where p^* and π^* denote the equilibrium price and profit respectively.

Using M_p^* to represent the market share of the proprietary software under the equilibrium price and M_o^* to denote that of the open source software, we have:

$$M_p^*=\frac{t-\gamma-k}{2[2(t-\gamma)-k]},\quad M_o^*=\frac{3t-3\gamma-k}{2[2(t-\gamma)-k]}.$$

The base-level qualities of proprietary and open source software are identical. However, the quality of the open source software increases with the number of users, and the price of open source software is zero. Hence, in equilibrium, the open source software has a bigger market share than the proprietary software.

4. Compatibility and Profits

In the basic model, we assumed that the open source and proprietary software were incompatible. However, the proprietary software producer may also choose for its product to be compatible to some degree with the open source software. Which is the best strategy? How does this compatibility decision change the profitability of proprietary software? To answer these questions, we extend our basic model to consider different degrees of compatibility. As in common with the Hotelling model, the analysis depends on whether the market is fully covered.

4.1. Fully Covered Market

We firstly present how the utility functions of the open source and proprietary software consumers change according to different compatibility strategy when the market is fully covered. Next, we summarize the net utilities of the open source and proprietary software consumers and the equilibrium outcomes under different compatibility strategies in Table 1.

- **Two-way compatibility**

Two-way compatibility is the case where the open source and proprietary software are compatible with each other. Then, users of the two software products share the same network externality.[5] Since the market is fully covered, the total number of software users is *1*. Thus, the network externality is $\gamma(q_o + q_p) = \gamma$.

- **Inward compatibility**

We define inward compatibility as the case where the proprietary software is compatible with the open source software, but the open source software is not compatible with the proprietary software. In this case, the network externality for users of the proprietary software is $\gamma(q_o + q_p) = \gamma$, and the network externality for users of the open source software is γq_o.

- **Outward compatibility**

Outward compatibility is defined as the case where the proprietary software is incompatible with the open source software, but the open source software is compatible with the proprietary software. In this case, the network externality for

[5] Following the previous literature (Farrell and Saloner 1992), we suppose the network externality intensity γ is the same for all the networks.

users of the proprietary software is γq_p, and the network externality for users of the open source software is $\gamma(q_o + q_p) = \gamma$.

Table 1 reports the net utilities of the open source and proprietary software consumers and the equilibrium outcomes under different compatibility strategies.[6]

	Two-way Compatibility	Inward Compatibility	Outward Compatibility	Incompatibility
U_o	$s + kq_o - tx + \gamma$	$s + kq_o - tx + \gamma q_o$	$s + kq_o - tx + \gamma$	$s + kq_o - tx + \gamma q_o$
U_p	$s - t(1-x) + \gamma - p$	$s - t(1-x) + \gamma - p$	$s - t(1-x) + \gamma q_p - p$	$s - t(1-x) + \gamma q_p - p$
$p*$	$\frac{t-k}{2}$	$\frac{t-k}{2}$	$\frac{t-\gamma-k}{2}$	$\frac{t-\gamma-k}{2}$
$\pi*$	$\frac{(t-k)^2}{4(2t-k)}$	$\frac{(t-k)^2}{4(2t-\gamma-k)}$	$\frac{(t-\gamma-k)^2}{4(2t-\gamma-k)}$	$\frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]}$
M_o^*	$\frac{3t-k}{2(2t-k)}$	$\frac{3t-2\gamma-k}{2(2t-\gamma-k)}$	$\frac{3t-\gamma-k}{2(2t-\gamma-k)}$	$\frac{3t-3\gamma-k}{2\left[2(t-\gamma)-k\right]}$
M_p^*	$\frac{t-k}{2(2t-k)}$	$\frac{t-k}{2(2t-\gamma-k)}$	$\frac{t-\gamma-k}{2(2t-\gamma-k)}$	$\frac{t-\gamma-k}{2[2(t-\gamma)-k]}$

Table 1: Fully Covered Market: Equilibrium Outcomes

4.2. Partly Covered Market

Suppose the market is partly covered. Then some consumers use neither the open source software nor the proprietary software. This may happen when the

[6] In this and following tables, U_o and U_p indicate the net utilities of open source and proprietary software consumers respectively; $p*$ and $\pi*$ are the equilibrium price and profit of the proprietary software; M_o^* and M_p^* represent the equilibrium market share of open source and proprietary software respectively.

benefit provided by the software is small compared with users' cost. The basic model is thus changed as follows:

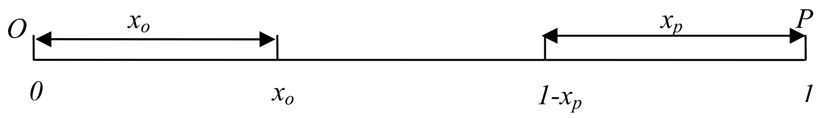

Figure 2: Competition: Partly Covered Market

The consumers at $x \in [0, x_o]$ would choose the open source software, where x_o is the location of the marginal consumer who is indifferent between using and not using the open source software. On the other hand, the consumers at $x \in [1 - x_p, 1]$ would choose the proprietary software, where $1 - x_p$ is the location of the marginal consumer who is indifferent between buying and not buying the proprietary software. The third group of consumers at $x \in [x_o, 1 - x_p]$ will choose neither the open source software nor the proprietary software.

The net utilities of the open source and proprietary software consumers under different compatibility strategies are reported in Table 2. By setting $U_o = 0$ and $U_p = 0$, we can get the equilibrium outcomes, which are also summarized in Table 2.

	Two-way Compatibility	Inward Compatibility	Outward Compatibility	Incompatibility
U_o	$s + kq_o - tx_o + \gamma(q_o + q_p)$	$s + kq_o - tx_o + \gamma q_o$	$s + kq_o - tx_o + \gamma(q_o + q_p)$	$s + kq_o - tx_o + \gamma q_o$
U_p	$s - tx_p + \gamma(q_o + q_p) - p$	$s - tx_p + \gamma(q_o + q_p) - p$	$s - tx_p + \gamma q_p - p$	$s - tx_p + \gamma q_p - p$
$p*$	$\frac{s(t-k)}{2(t-k-\gamma)}$	$\frac{s(t-k)}{2(t-k-\gamma)}$	$\frac{s}{2}$	$\frac{s}{2}$
$\pi*$	$\frac{(t-k)^2 s^2}{4(t^2 - 2t\gamma - tk + \gamma k)(t - \gamma - k)}$	$\frac{s^2 (t-k)^2}{4(t-\gamma)(t-\gamma-k)^2}$	$\frac{s^2}{4(t-\gamma)}$	$\frac{s^2}{4(t-\gamma)}$

M_o^*	$\frac{(2t^2-3t\gamma-2tk+\gamma k)s}{2(t^2-2t\gamma-tk+\gamma k)(t-\gamma-k)}$	$\frac{s}{t-\gamma-k}$	$\frac{s(2t-\gamma)}{2(t-\gamma)(t-\gamma-k)}$	$\frac{s}{t-\gamma-k}$
M_p^*	$\frac{(t-k)s}{2(t^2-2t\gamma-tk+\gamma k)}$	$\frac{s(t-k)}{2(t-\gamma)(t-\gamma-k)}$	$\frac{s}{2(t-\gamma)}$	$\frac{s}{2(t-\gamma)}$

Table 2: Partly Covered Market: Equilibrium Outcomes

5. Quality Differences

In section 4, the base-level qualities of the open source software and proprietary software are assumed to be equal. Since the open source software will increase in quality corresponding to the number of users, the market share of the open source software is higher than that of the proprietary software. In some situations, however, this condition may not be true. In this section, we extend the model and give the equilibrium outcomes when the base-level quality of the proprietary software is different from that of the open source software.

Suppose the open source software and the proprietary software products have different qualities, denoted by s_o and s_p respectively. With similar procedures in section 4, we can get the net utilities of open source and proprietary software consumers and the equilibrium outcomes under different market coverage conditions, which are reported in Table 3 and Table 4.

- **Fully Covered Market**

	Two-way Compatibility	Inward Compatibility	Outward Compatibility	Incompatibility
U_o	$s_o+kq_o-tx+\gamma$	$s_o+kq_o-tx+\gamma q_o$	$s_o+kq_o-tx+\gamma$	$s_o+kq_o-tx+\gamma q_o$
U_p	$s_p-t(1-x)+\gamma-p$	$s_p-t(1-x)+\gamma-p$	$s_p-t(1-x)+\gamma q_p-p$	$s_p-t(1-x)+\gamma q_p-p$

$p*$	$\frac{s_p - s_o + t - k}{2}$	$\frac{s_p - s_o + t - k}{2}$	$\frac{s_p - s_o + t - \gamma - k}{2}$	$\frac{s_p - s_o + t - \gamma - k}{2}$
$\pi*$	$\frac{(s_p - s_o + t - k)^2}{4(2t - k)}$	$\frac{(s_p - s_o + t - k)^2}{4(2t - \gamma - k)}$	$\frac{(s_p - s_o + t - \gamma - k)^2}{4(2t - \gamma - k)}$	$\frac{(s_p - s_o + t - \gamma - k)^2}{4[2(t - \gamma) - k]}$
M_o^*	$\frac{3t - k - s_p + s_o}{2(2t - k)}$	$\frac{3t - 2\gamma - k - s_p + s_o}{2(2t - \gamma - k)}$	$\frac{3t - \gamma - k - s_p + s_o}{2(2t - \gamma - k)}$	$\frac{3t - 3\gamma - k - s_p + s_o}{2[2(t - \gamma) - k]}$
M_p^*	$\frac{s_p - s_o + t - k}{2(2t - k)}$	$\frac{s_p - s_o + t - k}{2(2t - \gamma - k)}$	$\frac{s_p - s_o + t - \gamma - k}{2(2t - \gamma - k)}$	$\frac{s_p - s_o + t - \gamma - k}{2[2(t - \gamma) - k]}$

Table 3: Quality Differences: Fully Covered Market

- **Partly Covered Market**

	Two-way Compatibility	Inward Compatibility	Outward Compatibility	Incompatibility
U_o	$s_o + kq_o - tx_o + \gamma(q_o + q_p)$	$s_o + kq_o - tx_o + \gamma q_o$	$s_o + kq_o - tx_o$ $+ \gamma q_o + \gamma q_p$	$s_o + kq_o$ $- tx_o + \gamma q_o$
U_p	$s_p - tx_p + \gamma(q_o + q_p) - p$	$s_p - tx_p + \gamma(q_o + q_p) - p$	$s_p - tx_p$ $+ \gamma q_p - p$	$s_p - tx_p$ $+ \gamma q_p - p$
$p*$	$\frac{s_p}{2} + \frac{\gamma s_o}{2(t - k - \gamma)}$	$\frac{s_p}{2} + \frac{\gamma s_o}{2(t - k - \gamma)}$	$\frac{s_p}{2}$	$\frac{s_p}{2}$
$\pi*$	$\frac{[\gamma s_o + s_p(t - k - \gamma)]^2}{4[(t - \gamma)(t - \gamma - k) - \gamma^2](t - \gamma - k)}$	$\frac{(s_p t - s_p \gamma - s_p k + \gamma s_o)^2}{4(t - \gamma)(t - \gamma - k)^2}$	$\frac{s_p{}^2}{4(t - \gamma)}$	$\frac{s_p{}^2}{4(t - \gamma)}$

M_o^*	$\frac{\gamma^2 s_o}{2[(t-\gamma)(t-\gamma-k)-\gamma^2](t-k-\gamma)}+\frac{s_o}{t-k-\gamma}+\frac{\gamma s_p}{2[(t-\gamma)(t-\gamma-k)-\gamma^2]}$	$\frac{s_o}{t-k-\gamma}$	$\frac{\gamma s_p}{2(t-\gamma)(t-k-\gamma)}+\frac{s_o}{t-k-\gamma}$	$\frac{s_o}{t-k-\gamma}$
M_p^*	$\frac{\gamma s_o+s_p(t-k-\gamma)}{2[(t-\gamma)(t-\gamma-k)-\gamma^2]}$	$\frac{s_p(t-\gamma-k)+\gamma s_o}{2(t-\gamma)(t-\gamma-k)}$	$\frac{s_p}{2(t-\gamma)}$	$\frac{s_p}{2(t-\gamma)}$

Table 4: Quality Differences: Partly Covered Market

6. Best Compatibility Strategy

We examine the best compatibility strategy under different market coverage conditions by comparing the equilibrium outcomes in section 4 and section 5, and derive the following results:

Proposition 1 A proprietary software producer should never choose incompatibility or outward compatibility.

<proof> Please see the Appendix 1.

Under all conditions, the proprietary producer should not choose incompatibility or outward compatibility. Intuitively, to the proprietary software producer, inward compatibility can bring more profit than incompatibility by allowing proprietary software users to share the network benefits from open source users. Furthermore, incompatibility is always better than outward compatibility, in that it prevents open source software users to share the network benefits from proprietary software users. Therefore, incompatibility and outward compatibility would never be the best strategy. Consistent with these results, in reality, outward compatibility can hardly be observed. Incompatibility is also rare.

Proposition 2 When the market is partly covered, a proprietary software producer earns highest profit from two-way compatibility, followed by outward compatibility, and last, incompatibility or outward compatibility.

<proof> Please see the Appendix 1.

When the market is partly covered, one additional open source software user does not decrease the number of proprietary software users. In this case, an increase in open source software users can benefit proprietary software users through the network externality, without decreasing the market share of the proprietary software. Therefore, two-way compatibility is a win-win strategy: it increases the proprietary producers' profit while enhancing the user base of the open source software, which generate positive externalities for the users of the proprietary software.

Realistically, most software markets are not fully covered. Thus, two-way compatibility is common. Typical examples include: Internet vs. Mozilla, Microsoft Outlook vs. Mozilla Thunderbird. Furthermore, some proprietary software producers are improving the compatibility with open source software. For instance, Microsoft has committed in perpetuity to offering a royalty-free license of Office-related XML document formats, which encourages other open source software, such as Open Office, to create "filters" to read the files created in Microsoft Office (CNet News, 2004 b).

Proposition 3.1 When the market is fully covered, the proprietary software producer earns highest profit from inward compatibility, and lowest profit from outward compatibility. However, the profitability of two-way compatibility and incompatibility depends on the quality difference between the open source software and the proprietary software.

<proof> Please see the Appendix 1.

Proposition 3.2 When the market is fully covered and the quality difference between proprietary software and open source software is sufficiently small that $s_p - s_o < \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, the proprietary software producer earns highest

profit from inward compatibility, followed by two-way compatibility, incompatibility, and last, outward compatibility.

<proof> Please see the Appendix 1.

Proposition 3.3 When the market is fully covered and the quality difference between proprietary software and open source software is sufficiently large that $s_p - s_o > \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, the proprietary software producer earns highest profit from inward compatibility, followed by incompatibility, two-way compatibility, and last, outward compatibility.

<proof> Please see the Appendix 1.

A covered market indicates that the two products are in severe competition, i.e., one product can gain a user only if the other product loses one user. Thus, the proprietary software producer should adopt a strategy which promotes the increase of its product while restraining the growth of open source software. Inward compatibility allows proprietary software users to share the network externality of open source users, while preventing open source software users from sharing the network externality of the proprietary software. Therefore, it becomes the best compatibility strategy.

In a covered market, the profit ranking of the incompatibility and two-way compatibility varies according to the quality difference between the open source and proprietary software. On the one hand, the price of the proprietary software with two-way compatibility is higher than that with incompatibility (see Table 3). On the other hand, the market share of the proprietary software with two-way compatibility may be lower than that with incompatibility. This could happen when the proprietary software has sufficiently higher quality and becomes the dominating software in the market. Combining the price and market share factors together, we may conclude that once the quality of the proprietary software is high enough and the market share factor dominates the price factor, the proprietary software producer will choose incompatibility rather than two-way compatibility.

Such results provide theoretical explanation for the actual behavior of proprietary software producers. For instance, Windows update website denies access by Firefox, the new open source browser. It shows that Microsoft refuses to be compatible with Firefox, and thus restrains its growth.

The change of the profit ranking of incompatibility and two-way compatibility is consistent with the argument of Katz and Shapiro (1985). They develop a static model of oligopoly and conclude that the firms with large existing network externalities will tend to be against compatibility. In contrast, firms with small existing network externalities will tend to favor product compatibility. In the case of proprietary vs. open source software, when the proprietary software has sufficiently high quality, its existing network externalities are large. Hence, the proprietary software producer favors incompatibility over two-way compatibility. In contrast, when the quality of the proprietary software is sufficiently small, proprietary software has small existing network externalities. Therefore, two-way compatibility is more profitable than incompatibility.

Moreover, we extend the conclusion of Katz and Shapiro (1985) by adding the inward and outward compatibility, which they did not consider. We show that, in both fully covered and partly covered markets, inward compatibility is always a superior strategy to incompatibility. Hence, their conclusion that incompatibility can be the best strategy may not hold if inward compatibility can be realized.

7. Welfare

In this section, we calculate the social welfare and investigate how the parameters γ, s, k and t affect the social welfare. Social welfare in this model is the sum of sellers' profit and consumers' surplus. Since the open source software is freely distributed, we deem that sellers' profit from the open source software is zero. For tractability, we assume that the base-level qualities of proprietary and open source software are equal.

We investigate the case where the market is fully covered. The proprietary software producer is assumed to adopt the best strategy: inward compatibility. The welfare is denoted as *W* and presented as follows:

$$W = \int_{0}^{x_c} U_o dx + \int_{x_c}^{1} U_p dx + \pi_p$$

$$= \frac{(k+\gamma-t)(3t-2\gamma-k)^2}{4(2t-\gamma-k)^2} + s - \frac{(t-\gamma)(t-k)}{2(2t-\gamma-k)} + \frac{t}{2}$$

For the detailed calculations, please refer to Appendix 2.

We next investigate how the changes in the parameters *s*, *γ*, *k* and *t* affect the welfare. The results of the comparative statics are summarized in Table 5.[7]

	Increase in *s*	Increase in *γ*	Increase in *k*	Increase in *t*
Welfare	+	+	+	-

Table 5: Social Welfare

Firstly, it is intuitive that the increase in quality will raise the social welfare. Secondly, with the increase of the network externality intensity γ, the consumer surplus of both the open source users and the proprietary software users will increase. Thus, the social welfare will increase. Thirdly, higher *k* increases the quality of the open source software. Hence, consumer surplus increases. However, the increase in *k* reduces sellers' profit. From the comparative statics, it can be seen that the increase in consumer surplus dominates the decrease in sellers' profit, and welfare increases with the increase in *k*.

Moreover, the first order derivative of *W* with respect to *t* is:

$$\frac{\partial W}{\partial t} = \frac{(\gamma-k)(k+\gamma-t)(3t-2\gamma-k)}{2(2t-\gamma-k)^3} - \frac{(3t-2\gamma-k)^2-2(t-\gamma)(\gamma-k)}{4(2t-\gamma-k)^2} - \frac{(t-k)}{2(2t-\gamma-k)} + \frac{1}{2}.$$

[7] We also calculated welfare with two-way compatibility when the market is fully covered. The results of comparative statics are the same.

It is complicated to determine the sign of this equation analytically. Thus, we plot a figure to show how welfare changes according to *t*. In Figure 3, we fix $s = 0$ and $\gamma = 0.5$, and show that with the increase of taste difference *t*, the social welfare will decrease.[8] On the one hand, higher *t* binds consumers to the proprietary software and increases the monopoly power of the proprietary software. Thus, the sellers' profits will increase. On the other hand, higher *t* lowers the consumer surplus. From Figure 3, we can see that the latter impact dominates the former one and social welfare decreases with the increase in *t*.

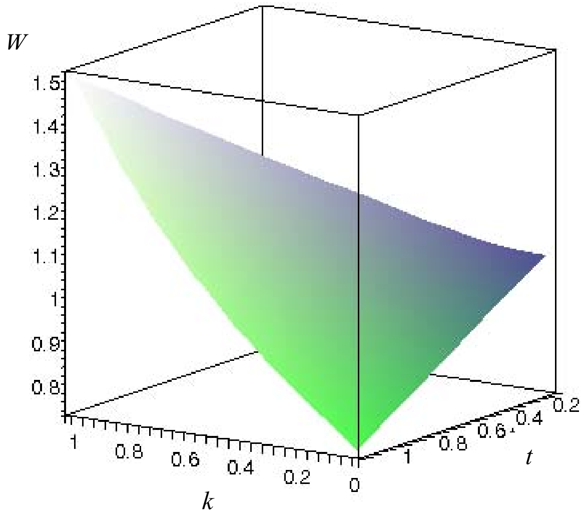

Figure 3: Welfare when $\gamma = 0.5$ *and* $s = 0$

8. Extension

8.1. Open Source Software Aiming at Increasing the Market Share

In the previous chapters, we have assumed that the proprietary software producer seeks profit maximization while open source software reacts passively. For certain

[8] We have tried other values of s and γ. The results are quite similar.

open source software, however, maximizing market share could be the aim. For instance, Sun uses StarOffice to battle with Miscrosoft Office and has won over 10 percent total market share (PC World News 2002).

In the above case, the competition changes to an asymmetric duopoly, where the proprietary software pursues the profit maximization and the open source software aims at maximizing market share. What would the best compatibility strategy be? We use a one-period game model to analyze this situation and derive the following proposition:

Proposition 4 When open source software aims at maximizing market share and proprietary pursues maximum profit, the competition between open source and proprietary software results in two-way compatibility in both fully covered and partly covered markets.

<proof> Please see the Appendix 3.

The different aims of the open source software will affect the compatibility choices of the proprietary software. As mentioned previously, when the market is partly covered, two-way compatibility is a win-win strategy for both open source and proprietary software, and therefore, a Nash Equilibrium. Moreover, when the market is fully covered and both parties have their maximization aims, if either party chooses incompatibility, the other will choose compatibility to share benefit through network externality. Therefore, two-way compatibility will be a Nash Equilibrium.

8.2. ‘Proprietary vs. Open’ and ‘Proprietary vs. Proprietary’

Today, opening the source code has become a trend that more and more proprietary software products are adopting. For instance, IBM has offered the source code of Cloudscape, a Java-based relational database software worth $85 million, to the Apache Software Foundation. The goal is to spur Java application development (IDG News Service 2004). Microsoft .Net, the rival of Java, will be seriously affected by such move of IBM.

How does the competition with open source software change the behavior of the proprietary software producer compared with the case where the rival is another proprietary producer? Will the change to open source software benefit the social welfare? We investigate these questions by comparing market outcomes of the basic model with those of the "proprietary vs. proprietary" case.

- **Comparison of price, market share and profit**

Let us consider a covered duopoly market where two proprietary software producers compete. We consider two symmetric cases: incompatibility and two-way compatibility. Here, we do not go through the details of the procedures because the results of two symmetric producers under Hotelling model are well known. We simply claim the following proposition.

Proposition 5 When the competitor is an open source software product, the proprietary software developer faces lower price, lower market share, and lower profit compared with the case when the rival is another proprietary software developer.

<proof> Please see the Appendix 4.

As Proposition 5 indicates, it is better for a proprietary software product to compete with another proprietary software product than contend with open source software. When its rival changes from a proprietary software product to open source software, the proprietary software producer is worse off in every aspect. Such results explain why Microsoft executives have publicly decried that the open-source movement is, at the minimum, bad for competition, and, at worst, a "cancer" to everything it touches (CNet News 2001). Furthermore, our results are consistent with Microsoft's sensitivity to Netscape or Sun opening up some parts of their products.

- **Comparison of the social welfare**

Proposition 6 When two proprietary software products are compatible and the market is fully covered, the change of one proprietary software to open source software will increase social welfare if and only if $t^3 < k(3t-k)^2$.

<proof> Please see the Appendix 5.

Such results are counterintuitive. Generally, it is believed that the open source software will benefit the social welfare by providing free high quality software. However, the results indicate that this is not always the case. The change of the proprietary software to open source software may lower social welfare under the condition that the consumers taste difference *t* is sufficiently high.

To get a better understand of the results, we draw Figure 4 and show how the value of $W_{op} - W_{pp} = \dfrac{-t^3 + k^3 + 9kt^2 - 6tk^2}{4(2t-k)^2}$ changes in term of *t* and *k*, where W_{pp} represents the social welfare with competition between two compatible proprietary software products and W_{op} indicates the social welfare when one of the proprietary software changes to open source software.

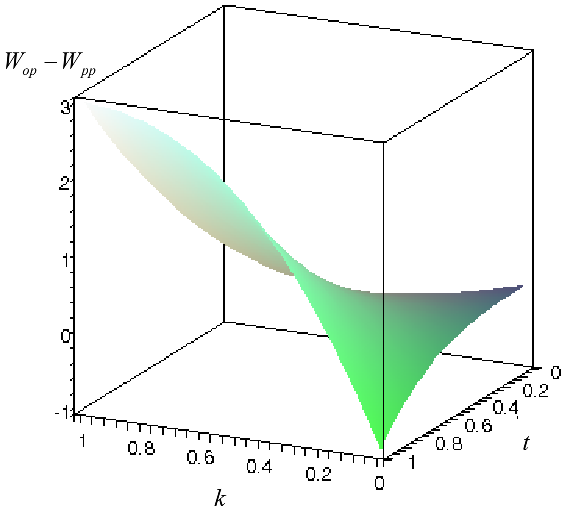

Figure 4: Picture of $W_{op} - W_{pp}$

From Figure 4, we can see that when *k* is small and *t* is large, the welfare of "proprietary vs. open" is smaller than that of "proprietary vs. proprietary", which means that social welfare is worse off when one of the proprietary software products changes to open source software. In contrast, when *k* is large and *t* is small, the change of the proprietary software to open source software benefits the social welfare.

The reason is that the change to the open source software increases the software quality, which has two opposite impacts on the social welfare. On the one hand, higher quality increases the consumer surplus and raises the social welfare. On the other hand, higher quality of the open source software drives the marginal consumers to switch from proprietary software to open source software. Since open source software covers more than half of the whole market share (see section 3), these marginal consumers move from the nearer software to the further one, and the consumer surplus decrease. In Figure 4, one can see that when t is large and k is small, the reduction in consumer surplus is the dominating factor and the social welfare will be lower.

Such results suggest that governments should not encourage the open source movement unconditionally. For instance, following Munich's decision to switch from Microsoft software to open source software, Paris postponed a similar move in light of the incompatible problem and high migration cost (ZDNet news, 2004).

9. Conclusion

We used the Hotelling model to investigate competition between open source and proprietary software. Firstly, we focused on the compatibility choices of the proprietary software. It was shown that the best compatibility strategy depends on the degrees of market coverage. When the market is fully covered, inward compatibility is the best strategy. When the market is partly covered, however, two-way compatibility is the best strategy. Moreover, the welfare analysis implied that the increase of t (the taste difference of consumers) may decrease the social welfare, although it can increase the profit of the proprietary software producer.

Next, we relaxed the conditions in the basic model and investigated the competition between open source and proprietary software in different scenarios. Firstly, we assumed that the open source software provider begins to maximize market share rather than reacts passively. The results showed that two-way compatibility is the best choice for both the proprietary and the open source software.

Secondly, we investigated the impact of the open source software from two aspects. From the aspect of a proprietary software producer, we found that it does not favor its proprietary rival changing to open source software. When the rival changes from proprietary to open source software, both the market share and the profit of the proprietary software producer will decease. From the aspect of social welfare, we found that when the rival of a proprietary software product changes from proprietary to open source software, social welfare may be lower off if the consumers taste difference is sufficiently high.

It is important to consider how the model's simplifying assumptions affect the conclusions. Firstly, we have assumed the compatibility strategies are chosen by the proprietary software producer. However, in the software market, the compatibility choice is decided not only by software producers but also by the software architecture. Sometimes, compatibility may not be feasible because the architecture of the two software products is quite different, while other times, the inward compatibility may be difficult to be technically implemented. Our results may not be applicable in such cases.

Secondly, we have assumed there is no installed user base for both open source and proprietary software. However, realistically, the open source software and the proprietary software may not enter the market simultaneously. The software which enters earlier will grab the consumers and have large existing network externalities. Therefore, the best compatibility strategy may change. Such limitation of the analysis suggests directions for future study. In the further work, it would be interesting to consider how the installed user base affects the compatibility choices of the proprietary software.

Finally, for tractability, we have assumed consumers taste difference is greater than the network externality intensity. If this assumption is violated, the equilibrium results are unstable. The switch of the marginal consumer from one software to the other results in that every consumer follows the switch. Such a tipping market needs further investigation.

References

Bessen, J. 2002. Open Source Software: Free Provision of a Complex Public Good. Working paper. http://www.researchoninnovation.org/opensrc.pdf.

Casadesus-Masanell, R., P. Ghemawat. 2003. Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows. Working paper. Harvard Business School.

CNet News 2001. *Why Microsoft Is Wary of Open Source*. http://news.com.com/2100-1001-268520.html?legacy=cnet.

CNet News. 2004a. *In Europe, Microsoft Aims to Mute Open-Source 'Noise'*. http://zdnet.com.com/2100-1104_2-5252994.html.

CNet News. 2004b. *Microsoft commits to XML docs for long term. http://techrepublic.com.com/5100-22_11-5442396.html.*.

Cremer, H., M. Marchand, J. Thisse. 1989. The public firm as an instrument for regulating an oligopolistic market. *Oxford Economic Papers.* 41 283-301.

Cremer, H., M. Marchand, . Thisse. 1991. Mixed oligopoly with differentiated products. *International Journal of Industrial Organization*. 9 43-53.

Dalle, J., N. Jullien. 2002. Open-source vs. proprietary software. Working paper. http://opensource.mit.edu/papers/dalle2.pdf.

DeFraja, G., F. Delbono. 1989. Alternative strategies of a public enterprise in oligopoly. *Oxford Economic Papers.* 41 302-311.

Fershtman, C. 1990. The interdependence between ownership status and market structure: The case of privatization. *Economica*. 57 319-328.

Fjell, K., D. Pal. 1996. A mixed oligopoly in the presence of foreign private firms. *Canadian Journal of Economics.* 29 737-743.

Halloween Documents. 1998. http://www.opensource.org/Halloween.

Hann, I-H., J. Roberts, S. Slaughter, R. Fielding. 2002. Economic incentives for participating in open source software projects. in *Proceedings of the Twenty-Third International Conference on Information Systems*. 365-372.

IDG News Service 2004. *IBM offers Cloudscape as open source code*. http://open.itworld.com/4923/040803cloudscape/page_1.html

International Data Corporation IDC. 2002. *Linux Market Share within Web Server Section to Grow.* http://thewhir.com/features/linux-market-share.cfm.

Johnson, J. P. 2001. Economics of Open Source Software. Working paper. Cornell University.

Katz, M., C. Shapiro. 1985. Network externalities, competition and compatibility. *American Economic Review.* 75(3) 424-440.

Katz, M., C. Shapiro. 1998. Antitrust in Software Markets. Working paper. University of California at Berkeley.

Kuan, J. 2001. Open Source Software as Consumer Integration Into Production. Working paper. University of California at Berkeley.

Lakhani, K. R., B. W. J. Bates, C. DiBona. 2002. *The Boston Consulting Group Hacker Survey.* http://www.osdn.com/bcg/BCGHACKERSURVEY-0.73.pdf.

Lakhani, K. R., R. G. Wolf. 2003. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Project. Working paper. MIT Sloan School of Management/The Boston Consulting Group. http://freesoftware.mit.edu/papers/lakhaniwolf.pdf

Lerner, J., J. Tirole. 2002. Some simple economics of open source. *The Journal of Industrial Economics*. 50(2), 197-234.

Mishra, B., A. Prasad, S. Raghunathan. 2002. Quality and profits under open source versus closed source. *in Proceedings of the Twenty-Third International Conference on Information Systems.* 349-363.

Netcraft Web Server Survey. 2004. *May 2004 Netcraft web server survey.* http://news.netcraft.com/archives/ 2004/05/index.html.

O'Reilly, T. 1999. Lessons from open-source software development. *Communications of the ACM.* 42(4) 33-37.

OSI. 2001. The Open Source Definition, http://opensource.org/docs/definition_plain.html.

PC World News 2002. *StarOffice Set to Challenge Microsoft's Office* http://www.pcworld.com/news/article/0,aid,99643,00.asp.

Raymond, E.S. 1999. The Cathedral and the Bazaar. Working paper. http://www.tuxedo.org/~esr/writings/cathedral-bazaar/ .

Vostroknutov, A. 2002. Economics of Open Source: A Brief Literature Overview. Working paper.

Weber, S. 2004. The success of open source. Harvard University Press.

White, M. 1996. Mixed oligopoly, privatization and subsidization. *Economics Letters.* 53 189-95.

ZDNet News 2004. *Paris shrugs off open-source plans*. http://news.zdnet.co.uk/software/linuxunix/0,39020390,39170365,00.htm.

Appendix 1. Best Compatibility Strategy

1.1. Fully Covered Market

In the proof procedures, strategy I denotes two-way compatible. Strategy II denotes inward compatibility. Strategy III and Strategy IV means the outward and incompatibility, respectively.

- **Equal base-level qualities**

Table 1 reports the equilibrium outcomes of fully covered market when open source and proprietary software have equal base-level qualities. From Table 1, we have:

$$p*(III) = p*(IV) < p*(I) = p*(II) \,, \qquad \text{(A-1)}$$

$$M_p^*(III) < M_p^*(IV) < M_p^*(I) < M_p^*(II) \,. \qquad \text{(A-2)}$$

By (A-1) and (A-2), it can be derived that $\pi*(III) < \pi*(IV) < \pi*(I) < \pi*(II)$.

- **Different base-level qualities**

When the market is fully covered and the base-level qualities of the open source and proprietary software are different, from Table 3, we have:

$$p(III) = p(IV) < p(I) = p(II) \,,$$

$$M_p^*(III) < M_p^*(IV) \,, M_p^*(I) < M_p^*(II) \,,$$

$$\pi_p^*(III) < \pi_p^*(IV) \,, \pi_p^*(I) < \pi_p^*(II) \,. \qquad \text{(A-3)}$$

To guarantee $M_p^*(IV) < 1$, we have

$$s_p - s_o < 3t - 3\gamma - k \,. \qquad \text{(A-4)}$$

Suppose $\pi_p^*(IV) = \pi_p^*(I)$, we have

$$s_p - s_o = \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2} \qquad \text{(A-5)}$$

We can get that :

If $s_p - s_o < \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, then $\pi_p^*(IV) < \pi_p^*(I)$. (A-6)

If $s_p - s_o > \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, then $\pi_p^*(IV) > \pi_p^*(I)$. (A-7)

Combining (A-6) with (A-3), it can be derived that:

If $s_p - s_o < \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, then $\pi_p^*(III) < \pi_p^*(IV) < \pi_p^*(I) < \pi_p^*(II)$.

Proposition 3.2 is proved.

Considering (A-7):

If $s_p - s_o > \frac{k + \sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, then $\pi_p^*(IV) > \pi_p^*(I)$. (A-7)

Under such case, to decide the sequence of the profit under different strategies, we need to compare $\pi_p^*(I)$ with $\pi_p^*(III)$, and compare $\pi_p^*(IV)$ with $\pi_p^*(II)$.

First, we compare $\pi_p^*(I)$ with $\pi_p^*(III)$:

Supposing $\pi_p^*(I) < \pi_p^*(III)$,

we have $s_p - s_o > t + \sqrt{(2t-k)(2t-k-\gamma)}$. (A-8)

Since $t+\sqrt{(2t-k)(2t-k-\gamma)} > 3t-\gamma-k > 3t-3\gamma-k$, (A-8) contradicts with (A-4). Therefore,

$$\pi^*(III) < \pi^*(I) . \qquad \text{(A-9)}$$

Secondly, we compare $\pi_p^*(IV)$ with $\pi_p^*(II)$:

Supposing $\pi_p^*(IV) > \pi_p^*(II)$, we can get that

$$s_p - s_o > t-\gamma+\sqrt{(2t-\gamma-k)(2t-\gamma-k-\gamma)} . \qquad \text{(A-10)}$$

Since $t-\gamma+\sqrt{(2t-\gamma-k)(2t-\gamma-k-\gamma)} > 3t-3\gamma-k$, (A-10) contradicts with (A-4). Therefore, it can be derived that

$$\pi^*(IV) < \pi^*(II) . \qquad \text{(A-11)}$$

Combining (A-3), (A-7), (A-9) and (A-11), it can be derived that

If $s_p - s_o > \dfrac{k+\sqrt{(2t-k)(2t-k-2\gamma)}}{2}$, then $\pi_p^*(IV) < \pi_p^*(I) < \pi_p^*(II) < \pi_p^*(III)$.

Proposition 3.3 is thus proved.

Proposition 3.2 and proposition 3.3 lead to proposition 3.1 spontaneously.

1.2. Partly Covered Market

- **Equal base-level qualities**

Table 2 report the equilibrium outcomes of partly covered market when open source and proprietary software have equal base-level qualities. From Table 2, we have:

$$p^*(III) = p^*(IV) < p^*(II) = p^*(I) , \qquad \text{(A-12)}$$

$$M_p^*(III) = M_p^*(IV) < M_p^*(II) < M_p^*(I)\,. \qquad \text{(A-13)}$$

By (A-12) and (A-13), it can be derived that $\pi^*(III) = \pi^*(IV) < \pi^*(II) < \pi^*(I)$.

- **Different base-level qualities**

Table 4 reports the equilibrium outcomes of partly covered market when open source and proprietary software have different base-level qualities. From Table 4, we have:

$$p^*(III) = p^*(IV) < p^*(II) = p^*(I)\,, \qquad \text{(A-14)}$$

$$M_p^*(III) = M_p^*(IV) < M_p^*(II) < M_p^*(I)\,. \qquad \text{(A-15)}$$

By (A-14) and (A-15), it can be derived that $\pi^*(III) = \pi^*(IV) < \pi^*(II) < \pi^*(I)$.

Such results are the same as those when open source and proprietary software products have the equal base-level quality. Therefore, the proposition 2 is proved.

Proposition 2 and proposition 3.1 prove proposition 1 spontaneously.

Appendix 2. Calculation of the Welfare

When the market is fully covered, proprietary software producer will choose inward compatibility. Under such scenario, the welfare denoted as W is calculated as:

$$\begin{aligned} W &= \int_{0}^{x_c} U_o dx + \int_{x_c}^{1} U_p dx + \pi_p \\ &= \int_{0}^{x_c} (s + kq_o - tx + \gamma q_o) dx + \int_{x_c}^{1} [s - t(1-x) + \gamma - p] dx + \pi_p \end{aligned} \qquad \text{(A-16)}$$

Substituting $q_o = \dfrac{3t - 2\gamma - k}{2(2t - \gamma - k)}$, $p = \dfrac{t-k}{2}$ and $x_c = \dfrac{3t - 2\gamma - k}{2(2t - \gamma - k)}$ into (A-16), we have:

$$\begin{aligned} W \;&=\; \int_{0}^{\frac{3t-2\gamma-k}{2(2t-\gamma-k)}} \left(s + k\frac{3t-2\gamma-k}{2(2t-\gamma-k)} - tx + \gamma\frac{3t-2\gamma-k}{2(2t-\gamma-k)}\right) dx \\ &\quad + \int_{\frac{3t-2\gamma-k}{2(2t-\gamma-k)}}^{1} [s - t(1-x) + \gamma - p] dx + \frac{(t-k)^2}{4(2t-\gamma-k)} \\ &= \frac{(k+\gamma-t)(3t-2\gamma-k)^2}{4(2t-\gamma-k)^2} + s - \frac{(t-\gamma)(t-k)}{2(2t-\gamma-k)} + \frac{t}{2} \end{aligned}$$

Appendix 3. Proof of Proposition 4

3.1. Fully Covered Market

Suppose the open source software aims at maximizing market share and the proprietary software pursues the maximum profits. And they choose the compatibility strategy simultaneously. The payoffs under each strategy are listed in Table A.1. The first number in each cell is the market share of the open source software, and the second number is the profit of the proprietary software.

Proprietary software

		Incompatibility	Compatibility
Open Source Software	Incompatibility	$M_o^* = \frac{3t-3\gamma-k}{2\left[2(t-\gamma)-k\right]}$, $\pi_p^* = \frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]}$	$M_o^* = \frac{3t-2\gamma-k}{2(2t-\gamma-k)}$, $\pi_p^* = \frac{(t-k)^2}{4(2t-\gamma-k)}$
	Compatibility	$M_o^* = \frac{3t+2\gamma-k}{2(2t+\gamma-k)}$, $\pi_p^* = \frac{(t-\gamma-k)^2}{4(2t-\gamma-k)}$	$M_o^* = \frac{3t-k}{2(2t-k)}$, $\pi_p^* = \frac{(t-k)^2}{4(2t-k)}$

Table A.1: Fully Covered Market: Competition Outcomes

The Nash equilibrium can be solved as (Compatible, Compatible). Therefore, when the market is fully covered, both open source and proprietary software will choose to be compatible with the rival and the result is two-way compatibility.

3.2. Partly Covered Market

Similarly, when the market is not fully covered, the payoffs under each strategy are listed in Table A.2.

Proprietary Software

		Incompatibility	Compatibility
Open Source Software	Incompatibility	$M_o^* = \frac{s}{t-\gamma-k}$, $\pi_p^* = \frac{s^2}{4(t-\gamma)}$	$M_o^* = \frac{s}{t-\gamma-k}$, $\pi_p^* = \frac{s^2(t-k)^2}{4(t-\gamma)(t-\gamma-k)^2}$
	Compatibility	$M_o^* = \frac{s(2t-\gamma)}{2(t-\gamma)(t-\gamma-k)}$, $\pi_p^* = \frac{s^2}{4(t-\gamma)}$	$M_o^* = \frac{(2t^2-3t\gamma-2tk+\gamma k)s}{2(t^2-2t\gamma-tk+\gamma k)(t-\gamma-k)}$, $\pi_p^* = \frac{(t-k)^2 s^2}{4(2t\gamma-\gamma k-t^2+tk)(\gamma+k-t)}$

Table A.2: Partly Covered Market: Competition Outcomes

The Nash equilibrium can be solved as (Compatible, Compatible). Therefore, when the market is not covered, both open source and proprietary software will choose to be compatible with their rival, and the result is two-way compatibility.

Appendix 4. Proof of Proposition 5

4.1. Incompatibility

When two software products are incompatible, the equilibrium outcomes of the proprietary software are listed in the following table, where the second row is the case of open vs. proprietary software and the third row is the case of proprietary vs. proprietary software.

Incompatibility	Equilibrium Price	Market Share	Profit
Case1: Open vs. Proprietary	$\frac{t-\gamma-k}{2}$	$\frac{t-\gamma-k}{2[2(t-\gamma)-k]}$	$\frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]}$
Case 2: Proprietary vs. Proprietary	$t-\gamma$	$\frac{1}{2}$	$\frac{t-\gamma}{2}$

Table A.3: The Comparison of Two Cases Where Two Products Are Incompatible

First, comparing the equilibrium price under two cases, it can be derived that:

$$\frac{t-\gamma-k}{2} < t-\gamma \,. \qquad \text{(A-17)}$$

Second, comparing the market share under two cases, it is obtained:

$$\frac{t-\gamma-k}{2[2(t-\gamma)-k]} < \frac{1}{2} \,. \qquad \text{(A-18)}$$

By (A-17) and (A-18), we can get:

$$\frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]} < \frac{t-\gamma}{2} \,.$$

Therefore, the equilibrium profit of the "proprietary vs. proprietary" case is higher than that of the "open vs. proprietary" case.

4.2. Two-way Compatibility

Table A.4 shows the equilibrium price, market share and the profit of the proprietary software when two products are two-way compatible.

Compatibility	Equilibrium Price	Market Share	Profit
Case1: Open vs. Proprietary	$\frac{t-k}{2}$	$\frac{t-k}{2(2t-k)}$	$\frac{(t-k)^2}{4(2t-k)}$
Case2: Proprietary vs. Proprietary	t	$\frac{1}{2}$	$\frac{t}{2}$

Table A.4: The Comparison of Two Cases Where Two Products Are Compatible

It is straightforward that the equilibrium profit under the "proprietary vs. proprietary" case is higher than that under "open vs. proprietary" case.

Appendix 5. Proof of Proposition 6

5.1. Social Welfare: Proprietary vs. Proprietary

Suppose that the market is fully covered and two software products are two-way compatible. The equilibrium price, market share and profits are shown as follows:

$$p_1^* = p_2^* = t \,,$$

$$\pi_1^* = \pi_2^* = \frac{t}{2} \,,$$

$$M_1^* = M_2^* = \frac{1}{2} \,,$$

where the footnote 1 and 2 denote the first proprietary and the second proprietary software respectively.

Using W_{pp} to denote the welfare when two proprietary software products compete together, we have:

$$\begin{aligned} W_{pp} &= \int_0^{\frac{1}{2}} U_1 dx + \int_{\frac{1}{2}}^{1} U_2 dx + \pi_1^* + \pi_2^* \\ &= \int_0^{\frac{1}{2}} (s - tx + \gamma - p) dx + \int_{\frac{1}{2}}^{1} [s - t(1-x) + \gamma - p] dx + t \qquad \text{(A-19)} \\ &= s + \gamma - \frac{t}{4} \end{aligned}$$

5.2. Social Welfare: Open vs. Proprietary

When two proprietary software compete together and one of them changes to open source software, we suppose that it aims at market share maximization. Therefore, the best strategy is two-way compatibility. Using W_{op} to denote the welfare, we have:

$$\begin{aligned} W_{op} &= \int_0^{x_c} U_o dx + \int_{x_c}^{1} U_p dx + \pi_p \\ &= \int_0^{x_c} (s + kq_o - tx + \gamma) dx + \int_{x_c}^{1} [s - t(1-x) + \gamma - p] dx + \pi_p \end{aligned} \tag{A-20}$$

By substituting $q_o = \frac{3t-k}{2(2t-k)}$, $p = \frac{t-k}{2}$ and $x_c = \frac{3t-k}{2(2t-k)}$ into (A-20), we have:

$$\begin{aligned} W_{op} &= \int_0^{\frac{3t-k}{2(2t-k)}} (s + k\frac{3t-k}{2(2t-k)} - tx + \gamma) dx + \int_{\frac{3t-k}{2(2t-k)}}^{1} [s - t(1-x) + \gamma - \frac{t-k}{2}] dx + \frac{(t-k)^2}{4(2t-k)} \\ &= s + \gamma - \frac{t(t-k)}{2(2t-k)} + \frac{(2k-t)(3t-k)^2 + t(7t-3k)(t-k)}{8(2t-k)^2} \end{aligned}$$

5.3. Social Welfare: "Open vs. Proprietary" vs. "Proprietary vs. Proprietary"

$$\begin{aligned} W_{op} - W_{pp} &= s + \gamma - \frac{t(t-k)}{2(2t-k)} + \frac{(2k-t)(3t-k)^2 + t(7t-3k)(t-k)}{8(2t-k)^2} - s - \gamma + \frac{t}{4} \\ &= \frac{-t^3 + k(3t-k)^2}{4(2t-k)^2} \end{aligned} \tag{A-21}$$

Therefore, we have:

If $t^3 > k(3t-k)^2$, then $W_{op} < W_{pp}$.

If $t^3 < k(3t-k)^2$, then $W_{op} > W_{pp}$