Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2007 Proceedings

European Conference on Information Systems (ECIS)

2007

Competition and Compatibility with Open Source Software

Zhaoli Meng
China Renmin University, mengzhao@ruc.edu.cn

Sang-Yong Tom Lee
Hanyang Universit

COMPETITION AND COMPATIBILITY WITH OPEN SOURCE SOFTWARE

Zhaoli Meng, China Renmin University, Beijing, mengzhao@ruc.edu.cn

Sang-Yong Tom Lee, Hanyang University

Abstract

We use Hotelling's linear city model to study asymmetric competition between open source and proprietary software, where only one party - the proprietary software producer - aims at maximizing the profit. We focus on the different compatibility choices of the proprietary software, which lead to different network externality, and thus result in different profit for proprietary software producers. It is found that the best compatibility strategy depends on the market coverage conditions: when the market is fully covered, one-way compatibility is the best choice for the proprietary software. On the other hand, when the market is not fully covered, two-way compatibility becomes the best strategy. These results are not affected by the software quality. Furthermore, once the open source software begins to purse the maximum market share, being two-way compatibility is the best strategy for both open source and proprietary software. We also investigate the impact of the open source software. It is shown that the proprietary software producer does not favor its proprietary rival changes to open source software. Moreover, such change may not benefit the social welfare.

Keywords: Open source software, Compatibility, Competition.

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 (O'Reilly 1999). One of the most famous and successful open source software projects is Linux, which commands a third market share within the web server sector and is expected to grow to 41 percent of the market 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, such as Perl language and Sendmail, have had significant market shares in their product categories (Lerner and Tirole 2002).

An interesting phenomenon is that facing open source competitors, proprietary software firms may adopt different strategies in compatibility[1]. In some cases, like Windows versus Linux, the proprietary software firm refuses to allow compatibility between its product and the open source software. But as one can see from Internet Explorer, a proprietary software product can be two-way compatible with an open source product such as Mozilla. Files created for IE users can be used without any difficulty by users of the Mozilla browser and vice versa. In other cases, a proprietary software firm may choose for its product to be one-way compatible, i.e., the files or programs designed for the open source software product can be used by the proprietary software, but the files or programs designed for the proprietary software may not be fully usable in the open source software.

Why would a proprietary software firm choose different strategies of compatibility in the face of competition with open source software? How would market factors affect the profitability of a proprietary software firm? Comparing with the case where the rival is another proprietary software product, how would the profit, price and market share of the proprietary software change in the case where the rival is open source software? Finally, what would be the welfare implications?

To answer these questions, we use Hotelling's linear city model to analyze the competition between an open source software product and a proprietary software product. Contrary to the traditional Hotelling model, where both firms pursue maximum profit, only one party – the proprietary software firm – tries to maximize its profit in our model. The open source software is passive because the price is set at zero and its quality depends on the total number of consumers. Such asymmetry requires obvious modification to the standard duopoly competition in the linear city model.

We find that when the market is covered, i.e., when all the consumers purchase one of the two products, one-way compatibility is the best strategy. Two-way compatibility is the second best strategy while incompatibility brings the lowest profit. However, when the market is not covered, two-way compatibility becomes the best strategy. We also find that the price, market share and profit for the proprietary software firm would be lower when the competitor is open source software than when the competitor is another proprietary software product. Finally, we illustrate that contrary to general belief, social welfare decreases with an increase in network externality.

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). Another theme of prior research focuses on the quality of open source software and the competition between open and proprietary software. An important conclusion is that open source software is not necessarily inferior in quality to proprietary software (Mishra et al. 2002, Kuan 2001, Johnson 2001, and Bessen 2002).Within this research theme, one strand has been to consider the competition between open and proprietary software. Casadesus-

[1] 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".

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. They showed that as long as Windows' pricing decision was not myopic, the results 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. First, the basic 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. Second, we suppose consumers have heterogeneous preferences for products, which are caused by two factors: 1) the learning cost of adopting a software product varies for different consumers, 2) because of the difference in their past experiences, the extent of 'lock-in' is different among consumers and their switching costs are different. The details of the model are present in section 3.

3 THE BASIC MODEL

Consider a software market where two software products with same quality *s* are located at the ends of a unit line. The open source software (O) is located at $x=0$ and the proprietary software (P) is located at $x=1$. Consumers are uniformly distributed along the unit interval. They differ in their preferences for the products, which is captured by their different locations. Specifically, for the 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 $t(1-x)$ if she chooses the proprietary software.

In the basic model, we assume that the two products are incompatible and the market is 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 last component γq_o refers to network externality. Following Katz and Shapiro (1985), network externalities are linear to the number of users who adopt the same or compatible software products. 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 and proprietary software products, then from $U_o = U_p$, we have:

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

The profit for the proprietary software firm π_p 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, we get:

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

The comparative statics presents three main results. First, higher t prompts consumers to stick to the nearest product. Thus, the profitabity of the proprietary software will increase. Second, higher k causes the quality of the open source software to increase, and it naturally decreases the profitability of the proprietary software product. Third, and most interestingly, higher intensity of network externality is disadvantageous to the proprietary software. Since the market share of the open source software is bigger than that of the proprietary software ($M_p^* < \frac{1}{2}$), the benefit of adopting the open source software is bigger than that of the proprietary software for users near the cutoff point. Thus, higher γ drives the potential consumer to deviate from the proprietary software.

4 COMPATIBILITY AND PROFITS

In the basic model, we have assumed the open and proprietary software products to be incompatible. However, proprietary software firms may also choose for their products to be two-way compatible or one-way compatible with open source software. How does this compatibility decision change the profitability of proprietary software? To answer this question, in this section, we extend our basic model to different settings of compatibility as well as to different market coverage conditions. And we list the results in the following Tables.

Market is covered	p^*	M_p^*	π^*
Incompatibility	$p^* = \frac{t-\gamma-k}{2}$	$M_p^* = \frac{t-\gamma-k}{2[2(t-\gamma)-k]}$	$\pi^* = \frac{(t-\gamma-k)^2}{4[2(t-\gamma)-k]}$
Compatibility	$p^* = \frac{t-k}{2}$	$M_p^* = \frac{t-k}{2(2t-k)}$	$\pi^* = \frac{(t-k)^2}{4(2t-k)}$
One way compatibility	$p^* = \frac{t-k}{2}$	$M_p^* = \frac{t-k}{2(2t-\gamma-k)}$	$\pi^* = \frac{(t-k)^2}{4(2t-\gamma-k)}$

Table 1. The profit when market is fully covered

Market is not covered	p^*	M_p^*	π^*
Incompatibility	$p^* = \frac{s}{2}$	$M_p^* = \frac{s}{2(t-\gamma)}$	$\pi^* = \frac{s^2}{4(t-\gamma)}$
Compatibility	$p^* = \frac{s}{2} + \frac{\gamma s}{2(t-k-\gamma)}$	$M^* = \frac{(t-k)s}{2(t^2-2t\gamma+\gamma k-tk)}$	$\pi^* = \frac{(t-k)^2 s^2}{4(2t\gamma-\gamma k-t^2+tk)(\gamma+k-t)}$
One way compatibility	$p^* = \frac{s}{2} + \frac{s\gamma}{2(t-k-\gamma)}$	$M_p^* = \frac{(t-k)s}{2(t^2-2t\gamma+\gamma k-tk)}$	$\pi^* = \frac{s^2(t-k)^2}{4(t-\gamma)(t-\gamma-k)^2}$

Table2. The profit when market is not fully covered

Table 1 lists the proprietary software firm's optimal prices, market shares and profit with different choices of compatibility when the market is covered. Table 2 lists those when the market is not covered. By comparing the profit of the firm under different compatibility conditions, we claim the following proposition.

Proposition 1 When the market is covered, one-way compatibility brings the highest profit to the proprietary software firm followed by two-way compatibility and incompatibility. However, when the market is not covered, two-way compatibility becomes the best strategy for the proprietary software firm.

This result implies that the best compatibility strategy depends on market coverage conditions. When the market is covered, the two products are under severe competition, i.e., one more user of one product always comes with one less consumer of the other product. One-way 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 strategy.

In contrast, when the market is not covered, one additional open source software user does not decrease the number of proprietary software users. In this case, an increase in open source users can benefit proprietary software users through network externality without hurting the market share of the proprietary software. Therefore, two-way compatibility becomes a win-win strategy which can increase the profit of the proprietary software while enhancing the market share of the open source software.

5 'PROPRIETARY VERSUS OPEN' AND 'PROPRIETARY VERSUS PROPRIETARY'

Today, more and more proprietary software products have to compete with open source software products. How does the competition with open source software change the behavior of the proprietary software firm compared to the case where the rival is another such firm? In this section, we compare the optimal prices, market shares and profit when the rival changes from a proprietary software firm to an open source software producer.

Let us consider a covered duopoly market where two symmetric proprietary software firms compete. Here, we do not go through the details of the procedure because the results of two symmetric firms under Hotelling's model are well known. We simply claim the following proposition.

Proposition 2 Regardless of compatibility strategies, when the competitor is an open source software product, the proprietary software firm faces lower price, lower market share, and lower profit than when the rival is another proprietary software firm.

As Proposition 2 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 firm is hurt in every aspect.

6 WELFARE

In this section, we calculate welfare using the basic model and investigate how the parameters γ, s, k and t affect social welfare. Social welfare in this model is the sum of sellers' profit and consumers' surplus, which is presented as follows.

$$W = \int_0^{x_c} U_o dx + \int_{x_c}^{1} U_p dx + \pi_p = \frac{(3t-3\gamma-k)^2(2k-3t+4\gamma)}{8[2(t-\gamma)-k]^2} + s - \frac{2t-3\gamma-k}{2} + \frac{(3t-3\gamma-k)(3t-4\gamma-k)}{2[2(t-\gamma)-k]}$$

Since $\frac{\partial W}{\partial s} = 1$, it can be derived that welfare increases according to the increase in software quality s, which is too intuitive to be interesting. However, other comparative statics results are very complicated and not easy to be analyzed. Thus, we plot following figures to show how welfare changes according to each parameter while assuming $s = 0$.

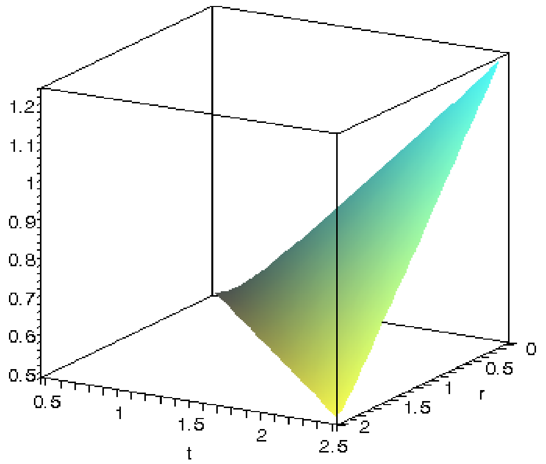

Figure 1. *Welfare changes with respect to t and* γ.

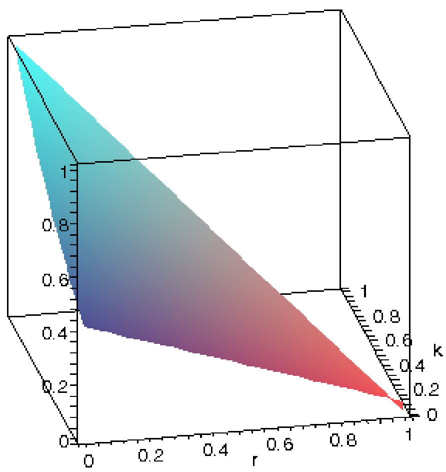

Figure 2. *Welfare changes with respect to k and* γ.

Observation 1: Figure 1 presents that welfare increases with the increase in t and welfare decreases with the increase in γ .

Observation 2: Figure 2 shows that welfare increases with the increase in k and welfare decreases with the increase in γ .

It is interesting that increase in network externality intensity γ decreases welfare. As mentioned in section 3, the market share of the open source software is bigger than that of the proprietary software. Therefore, increase in γ leads potential consumers to move from the proprietary software to the open source software, and thus deceases the profit of the proprietary software producer. Although network externality increases because some consumers move from the smaller network to the bigger one, the figures show that the increase in network externality cannot compensate the decrease in sellers' profit and welfare decreases as γ increases.

Welfare increases with the increase in utility cost t. This is because t binds consumers to the proprietary software. Therefore, the monopoly power of sellers of the proprietary software will increase. As a result, sellers' profits and welfare will increase. The increase of k enhances the quality of the open source software increases. Therefore, the consumer surplus and the welfare increases.

7 CONCLUSION AND FUTURE WORK

We have used Hotelling's model to investigate the competition between open and proprietary software. By comparing optimal price, market share and maximum profit under each strategic choice in compatibility, our study suggests that the best strategy for the proprietary software depends on whether the market is covered. When the market is covered, one-way compatibility is the best strategy for the proprietary software firm. When the market is not covered, however, two-way compatibility is the best strategy as it can increase both the profit of the proprietary software and the market share of the open source software. It is also shown that the proprietary software firm does not favor having an open source software product as a rival in the market. The welfare results imply that increase in network externality intensity γ lowers welfare. In contrast, increase in unit utility cost t increases welfare.

The limitations of our analysis suggest several directions for future research. First, we have assumed that the open source software equals the proprietary software in quality and the open source software will increase in quality according to the number of users. Therefore, the market share of the open source software is higher than that of the proprietary software. In further work, it would be interesting to consider what happens once the market share of the proprietary software exceeds that of the open source software. Second, we have assumed that the proprietary software firm seeks profit maximization while the producer of the open source software passively reacts to the price set by the proprietary software firm. For certain open source software, however, maximizing market share could be the aim. Such case can be modeled in future work. Finally, for tractability, we have employed a one-period model and assumed that the compatibility choice is on the proprietary software firm. In future work, a two-stage model could be concerned, where the open source software producer may change the compatibility strategy to win the maximum market share.

References

Bessen, J. 2002. Open Source Software: Free Provision of a Complex Public Good. Working paper.

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

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. UC Berkeley.

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

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

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