ISSN : 0974 - 7435 Volume 10 Issue 12

2014

BioTechnology
An Indian Journal

FULL PAPER
BTAIJ, 10(12), 2014 [6348-6356]

The impact of open source software on proprietary software firms' profit and social welfare

Mingqing Xing
[1]School of Economics and Management, Weifang University, Weifang 261061, (CHINA)
[2]Neural Decision Lab, Weifang University, Weifang 261061, (CHINA)
E-mail : mqxing1979@163.com

ABSTRACT

open source software has been achieved notable success in recent years and becomes a powerful rival to proprietary software in the software industry. Through modifying the Cournot model, this study analyzes how open source software affects the profit of proprietary software firms and social welfare. This paper supposes that proprietary software firms aim at maximizing profit and open source software can be freely available. It mainly finds that the emergence of open source software doesn't always decrease (resp. reduce) the proprietary software firm's profit or output (resp. price) and increase the social welfare. This conclusion contradicts the traditional recognition of people to open source software.

KEYWORDS

Open source software; Proprietary software; Social welfare; Network externality.

INTRODUCTION

Open source software (OSS) is typically developed by volunteers from around the global and becomes a powerful rival to proprietary software (PS) in many software markets in recent years. In server operating system market, the open source Linux operating system commands about 30 percent share, where Microsoft's Windows, a proprietary software, holds approximately 50 percent share (Netcraft, 2001)[1]; More than 60 percent of websites use the open source web server software Apache in web server market, while only less than 30 percent supports Microsoft's Internet Information Services (a proprietary software) (Netcraft, 2006)[2]; Sendmail, as an open source software, holds about 80 percent share in e-mail traffic market (Weber, 2004)[3].

According to O'Reilly (1999)[4], open source software is software whose source code allows software developers to share, identify and correct errors, and redistribute, which is usually available at no charge, and which is often developed by voluntary efforts. The academic literature pays more and more attention to the open source problem, in which competition between open source and proprietary software is a very hot area. Dalle and Jullien (2001)[5] investigate the technological competition between open source and proprietary software; Meng and Lee (2005)[6] and Xing (2010)[7] consider the compatibility of proprietary software to open source software; Mustonen (2005)[8] analyzes competition between proprietary and open source software when proprietary software firms support the development of substitute open source software; Lin (2008)[9] examines the influence of user skill and network effect on the software market where proprietary software competes with open source software; Xing (2012)[10] studies how open source software affects the quality of proprietary software. From the views of technology competition, software compatibility, user skill and software quality, the above literature analyzes competition between open source and proprietary software. However most of them have not considered how open source software influences the profit of proprietary software firms and social welfare. Though extending the Cournot model, this paper studies the impact of open source software on proprietary software firms and social welfare in a software market with network externalities. We know that the software market generally presents network externalities, which is that the benefit that users enjoy from buying one or several of its products depends on the number of other users that use the same or compatible products (Katz and Shapiro, 1985)[11].

The rest of this study is organized as follows. In section 2, two models are presented. In section 3, the optimal results are compared. In final part, the paper is concluded.

THE MODELS

To analyze the impact of open source software on the profit of proprietary software firm and social welfare, this paper sets up two models. One model supposes that the proprietary software firm is a monopoly in the market. The other one supposes that the proprietary software firm faces competition from open source software.

The model of proprietary software monopolizing

This part considers only proprietary software in a software market, which is noted by subscript 'p'. The proprietary software is produced by a proprietary software firm. Extending the Cournot model as in[12], the inverse demand function for proprietary software is given by

$$p_p = a_p + u_p(q_p^0) - q_p, \quad \textbf{(1)}$$

where $a_p > 0$ and $u_p(q_p^0) \ge 0$. In (1), a_p denotes the reservation price of proprietary software; $u_p(q_p^0)$ measures the network externalities on the demand function (Regibeau and Rockett, 1996)[13], in which q_p^0 is the initial network size for proprietary software firm.

According to (1), the profit function for proprietary software firm is given by

$$\pi_p = p_p q_p = (a_p + u_p - q_p)q_p \text{.} \qquad \textbf{(2)}$$

Note that the marginal cost of proprietary software is assumed to equal zero.

Taking the derivative of (2) with respect to q_p, and then setting it equal to zero, the first order condition is given by $\frac{\partial \pi_p}{\partial q_p} = a_p + u_p - 2q_p = 0$. Therefore, the optimal quantity for proprietary software firm is obtained

$$q_p^* = \frac{a_p + u_p}{2} \text{.} \qquad \textbf{(3)}$$

Obviously, q_p^* satisfies the second order condition ($\frac{\partial^2 \pi_p}{\partial q_p^2} = -2 < 0$), therefore it is the unique optimal solution.

Substituting (3) into (1) and (2), the optimal price and profit for proprietary software firm are respectively given by

$$p_p^* = \frac{a_p + u_p}{2} \text{,} \qquad \textbf{(4)}$$

$$\pi_p^* = \frac{(a_p + u_p)^2}{4} \text{.} \qquad \textbf{(5)}$$

The social welfare in this model equals consumers' surplus plus sellers' profit. Given the inverse demand function in (1), the social welfare at the optimal solution meeting the first order condition can be calculated

$$\begin{aligned} SW &= \int_0^{q_p^*} (a_p + u_p - x)dx \\ &= \pi_p^* + \frac{(q_p^*)^2}{2} = \frac{3}{8}(a_p + u_p)^2 \end{aligned} \text{.} \qquad \textbf{(6)}$$

The model of proprietary software competing with open source software

Consider two software producers in the market in this part. One produces proprietary software and the other one produces open source software. Open source software is denoted by subscript 'o'. The inverse demand functions for proprietary and open source software are respectively given by

$$p_p = a_p + u_p^{'}(q_p^0, q_o^0) - q_p - dq_o \text{,} \qquad \textbf{(7)}$$

$$p_o = a_o + u_o^{'}(q_p^0, q_o^0) + \beta q_o - q_o - dq_p - c \text{,} \qquad \textbf{(8)}$$

where $a_o > 0$, $a_p > 0$, $q_o^0 \ge 0$, $q_p^0 \ge 0$, $0 < \beta < 1$, $0 < d < 1$ and $0 < c < a_o$. In (7) and (8), a_p and a_o denote the reservation prices for proprietary and open source software respectively; d measures the degree of substitution between proprietary and open source software; $u_p^{'}(q_p^0, q_o^0)$ and $u_o^{'}(q_p^0, q_o^0)$ refer to the network externalities on the demand function for proprietary and open source software respectively, in which q_i^0 ($i = p, o$) is the initial network scale for firm i ($i = p, o$); c denotes the learning (maintenance or development) costs when consumers use open source software; β indicates the contribution degree of each user to the reservation price when he/she uses open source software (or call it user's software development skill parameter).

In reality, open source software is usually free and can be downloaded in open source community. Therefore, p_o is assumed to equal 0 in this study. Solving (8), we obtain

$$q_o = \frac{a_o + u_o^{'} - dq_p - c}{1-\beta}. \quad \textbf{(9)}$$

Substituting (9) into (7), we obtain the price for proprietary software

$$p_p = \frac{(1-\beta)(a_p + u_p^{'}) - d(a_o + u_o^{'} - c) - (1-\beta-d^2)q_p}{1-\beta}. \quad \textbf{(10)}$$

According to (9) and (10), the profit functions for proprietary and open source software producer are respectively given by

$$\pi_p = p_p q_p = \frac{\begin{pmatrix}(1-\beta)(a_p + u_p^{'}) - d(a_o + u_o^{'} \\ -c) - (1-\beta-d^2)q_p\end{pmatrix} q_p}{1-\beta}, \quad \textbf{(11)}$$

$$\pi_o = p_o q_o = 0. \quad \textbf{(12)}$$

Note that the marginal costs for proprietary and open source software are assumed to equal zero.

As the profit for open source software firm always equals zero, we only need to solve the optimal quantity for proprietary software firm. Taking the derivative of (11) with respect to q_p, and then setting it equal to zero, we derive

$$\frac{\partial \pi_p}{\partial q_p} = \frac{\begin{pmatrix}(1-\beta)(a_p + u_p^{'}) - d(a_o + u_o^{'} \\ -c) - 2(1-\beta-d^2)q_p\end{pmatrix}}{1-\beta} = 0. \quad \textbf{(13)}$$

Solving (13), the optimal quantity for proprietary software firm is given by

$$q_p^{'*} = \frac{(1-\beta)(a_p + u_p^{'}) - d(a_o + u_o^{'} - c)}{2(1-\beta-d^2)}. \quad \textbf{(14)}$$

This paper supposes the parameters meet the inequality $q_p^{'*} > 0$. To make sure $q_p^{'*}$ the unique optimal solution, $q_p^{'*}$ must satisfy the second order condition, which requires

$$\frac{\partial^2 \pi_p}{\partial q_p^2} = \frac{-2(1-\beta-d^2)}{1-\beta} < 0. \quad \textbf{(15)}$$

Since $1-\beta>0$, the (15) holds if

$$1-\beta-d^2>0\,. \quad \textbf{(16)}$$

The parameters are assumed to meet the above inequality, therefore $q_p^{'*}$ is the unique optimal quantity for proprietary software firm.

Substituting (14) into (9), (10) and (11), the optimal price and profit for proprietary software firm and the optimal quantity for open source software firm are respectively given by

$$p_p^{'*}=\frac{a_p+u_p^{'}}{2}-\frac{(a_o+u_o^{'}-c)d}{2(1-\beta)}, \quad \textbf{(17)}$$

$$\pi_p^{'*}=\frac{[(1-\beta)(a_p+u_p^{'})-d(a_o+u_o^{'}-c)]^2}{4(1-\beta)(1-\beta-d^2)}, \quad \textbf{(18)}$$

$$q_o^{'*}=\frac{\begin{pmatrix}[2(1-\beta)-d^2](a_o+u_o^{'}-c)\\ -(1-\beta)d(a_p+u_p^{'})\end{pmatrix}}{2(1-\beta)(1-\beta-d^2)}\,. \quad \textbf{(19)}$$

The social welfare is given by

$$\begin{aligned}SW'&=\int_0^{q_p^{'*}}(a_p+u_p^{'}-x-dq_o^{'*})dx\\ &\quad+\int_0^{q_o^{'*}}(a_o+u_o^{'}+\beta q_o^{'*}-x-dq_p^{'*}-c)dx\\ &=\frac{\begin{pmatrix}(1-\beta)[3(1-\beta)-2d^2][(1-\beta)(a_p+u_p^{'})\\ -d(a_o+u_o^{'}-c)]^2+\{[2(1-\beta)-d^2](a_o\\ +u_o^{'}-c)-(1-\beta)d(a_p+u_p^{'})\}^2\end{pmatrix}}{8(1-\beta)^2(1-\beta-d^2)^2}\end{aligned}\,. \quad \textbf{(20)}$$

COPARISON

The optimal price, quantity and profit for proprietary software firm and the social welfare in aforementioned models are compared in this section. For purposes of analysis, this section only considers the network externalities on demand are linear (Katz and Shapiro, 1985)[11] and the intensity of network externalities, the reservation price and initial network scale for proprietary software firm equal in two models in section 2. The network externality functions mentioned in section 2 are respectively given by

$$u_p(q_p^0)=\alpha q_p^0, \quad \textbf{(21)}$$

$$u_p^{'}(q_p^0,q_o^0)=\alpha(q_p^0+k_1q_o^0), \quad \textbf{(22)}$$

$$u_o^{'}(q_p^0,q_o^0)=\alpha(q_o^0+k_2q_p^0), \quad \textbf{(23)}$$

where $q_p^0 \geq 0$, $q_o^0 \geq 0$, $\alpha \geq 0$, $0 \leq k_1 \leq 1$ and $0 \leq k_2 \leq 1$. The parameter α is the intensity of network externalities, k_1 is the compatibility degree of proprietary software to open source software and k_2 is the compatibility degree of open source software to proprietary software.

Comparison of prices, quantities and profits

Setting $c_1 = a_o + \alpha(q_o^0 + k_2 q_p^0) - \alpha k_1 (1-\beta) q_o^0 / d$, $c_2 = a_o + \alpha(q_o^0 + k_2 q_p^0) + d(a_p + \alpha q_p^0) - \alpha k_1 (1-\beta) q$ and $c_3 = a_o + \alpha(q_o^0 + k_2 q_p^0) - \left(\begin{array}{c} [a_p + \alpha(q_p^0 + k_1 q_o^0)] \times \\ (1-\beta) - (a_p + \alpha q_p^0) \times \\ \sqrt{(1-\beta)(1-\beta-d^2)} \end{array} \right) \Big/ d$, the following proposition is obtained.

Proposition 1: (i) when $c < c_1$, $p_p^* > p_p^{'*}$; when $c > c_1$, $p_p^* < p_p^{'*}$; (ii) when $c < c_2$, $q_p^* > q_p^{'*}$; when $c > c_2$, $q_p^* < q_p^{'*}$; (iii) when $c < c_3$, $\pi_p^* > \pi_p^{'*}$; when $c > c_3$, $\pi_p^* < \pi_p^{'*}$.

Proof. (i) According to (4) and (17), $p_p^* - p_p^{'*} = \frac{[a_o + \alpha(q_o^0 + k_2 q_p^0)]d - \alpha k_1 (1-\beta) q_o^0 - cd}{2(1-\beta)}$. Therefore, $p_p^* > p_p^{'*}$ when $c < c_1$ and $p_p^* < p_p^{'*}$ when $c > c_1$; (ii) According to (3) and (14), $q_p^* - q_p^{'*} = \frac{\left(\begin{array}{c} [a_o + \alpha(q_o^0 + k_2 q_p^0)]d + (a_p + \\ \alpha q_p^0) d^2 - \alpha k_1 (1-\beta) q_o^0 - cd \end{array} \right)}{2(1-\beta)}$.

Therefore, $q_p^* > q_p^{'*}$ when $c < c_2$ and $q_p^* < q_p^{'*}$ when $c > c_2$; (iii) According to (5) and (18), $\sqrt{\pi_p^*} - \sqrt{\pi_p^{'*}} = \frac{\left(\begin{array}{l} 2(a_p + \alpha q_p^0)\sqrt{(1-\beta)(1-\beta-d^2)} \\ -[a_p + \alpha(q_p^0 + k_1 q_o^0)](1-\beta) \\ +[a_o + \alpha(q_o^0 + k_2 q_p^0)d - dc] \end{array} \right)}{2\sqrt{(1-\beta)(1-\beta-d^2)}}$. Therefore, $\sqrt{\pi_p^*} > \sqrt{\pi_p^{'*}}$ when $c < c_3$ and $\sqrt{\pi_p^*} < \sqrt{\pi_p^{'*}}$ when $c > c_3$.

We obtain $\pi_p^* > \pi_p^{'*}$ when $c < c_3$ and $\pi_p^* < \pi_p^{'*}$ when $c > c_3$. □

Proposition 1 shows that, if the learning (maintenance or development) costs of open source software are sufficiently high, the optimal quantity and profit (resp. price) for proprietary software firm are more (resp. higher) when proprietary software monopolizes the market than when it competes with open source software, and the opposite situations may appear if the learning (maintenance or development) costs of open source software are sufficiently low. That is, the appearance of open source software in a software market does not necessarily decrease (resp. lower) the quantity and profit (resp. price) of proprietary software firm. This conclusion depends on the level of the learning (maintenance or development) costs for open source software. Proposition 2: when $k_1 = 0$, $p_p^* > p_p^{'*}$, $q_p^* > q_p^{'*}$ and $\pi_p^* > \pi_p^{'*}$.

Proof. When $k_1 = 0$, $c_1 = a_o + \alpha(q_o^0 + k_2 q_p^0) > c$ and $c_2 = a_o + \alpha(q_o^0 + k_2 q_p^0) + d(a_p + \alpha q_p^0) > c$. According to part (i) and (ii) of proposition 1, $p_p^* > p_p^{'*}$ and $q_p^* > q_p^{'*}$. Moreover, $\pi_p^* > \pi_p^{'*}$ when $k_1 = 0$ because $\pi_p^* = p_p^* q_p^*$ and $\pi_p^{'*} = p_p^{'*} q_p^{'*}$. □

Proposition 2 demonstrates that, if the compatibility degree of proprietary software to open source software equals zero (i.e. proprietary software isn’t compatible to open source software), proprietary software firm will price higher, output more and obtain greater profit when proprietary software monopolizes the market than when it competes with open source software. Notice that the results of proposition 2 don’t depend on the learning (maintenance or development) costs of open source software.

Making $prob\{A\}$ indicate the probability of event A happening, the following proposition can be obtained.

Proposition 3: (i) $prob\{p_p^* > p_p^{'*}\}$, $prob\{q_p^* > q_p^{'*}\}$ and $prob\{\pi_p^* > \pi_p^{'*}\}$ (resp. $prob\{p_p^* < p_p^{'*}\}$, $prob\{q_p^* < q_p^{'*}\}$ and $prob\{\pi_p^* < \pi_p^{'*}\}$) will not increase (resp. decrease) with c or k_1; (ii) $prob\{p_p^* > p_p^{'*}\}$ and $prob\{q_p^* > q_p^{'*}\}$ (resp. $prob\{p_p^* < p_p^{'*}\}$ and $prob\{q_p^* < q_p^{'*}\}$) will not decrease (resp. increase) with β, d, a_o, k_2 or q_p^0; (iii) when $k_1 = 0$, $prob\{\pi_p^* > \pi_p^{'*}\}$ (resp. $prob\{\pi_p^* < \pi_p^{'*}\}$) will not decrease (resp. increase) with β or d; (iv) when $k_1 = k_2 = 0$, $prob\{\pi_p^* > \pi_p^{'*}\}$ (resp. $prob\{\pi_p^* < \pi_p^{'*}\}$) will not decrease (resp. increase) with q_p^0.

Proof. (i) $prob\{c < c_1\}$ will not increase with c. Combining with proposition 1, $prob\{p_p^* > p_p^{'*}\}$, $prob\{q_p^* > q_p^{'*}\}$ and $prob\{\pi_p^* > \pi_p^{'*}\}$ (resp. $prob\{p_p^* < p_p^{'*}\}$, $prob\{q_p^* < q_p^{'*}\}$ and $prob\{\pi_p^* < \pi_p^{'*}\}$) will not increase (resp. decrease) with c. Since $\frac{\partial c_1}{\partial k_1} = -\frac{\alpha(1-\beta)q_o^0}{d} \le 0$, c_1 will not increase with k_1. Therefore, $prob\{c < c_1\}$ will not increase with k_1. According to proposition 1, $prob\{p_p^* > p_p^{'*}\}$, $prob\{q_p^* > q_p^{'*}\}$ and $prob\{\pi_p^* > \pi_p^{'*}\}$ (resp. $prob\{p_p^* < p_p^{'*}\}$, $prob\{q_p^* < q_p^{'*}\}$ and $prob\{\pi_p^* < \pi_p^{'*}\}$) will not increase (resp. decrease) with k_1; similarly to part (i), part (ii) and (iii) also can be proven. □

Proposition 3 shows that, the learning (maintenance or development) costs of open source software, user's development skill, initial network scale for proprietary software firm, software compatibility and software differentiation may influence the probability of $p_p^* > p_p^{'*}$, $q_p^* > q_p^{'*}$, $\pi_p^* > \pi_p^{'*}$, $p_p^* < p_p^{'*}$, $q_p^* > q_p^{'*}$ or $\pi_p^* < \pi_p^{'*}$.

Comparison of social welfare

This part compares the social welfare levels. According to (6) and (20), the social welfare difference in two cases is

$$SW - SW' = \frac{\begin{pmatrix} 3(1-\beta)^2(1-\beta-d^2)^2(a_p \\ +u_p)^2 - (1-\beta)[3(1-\beta) \\ -2d^2][(1-\beta)(a_p+u_p^{'}) \\ -d(a_o+u_o^{'}-c)]^2 - \{[2(1 \\ -\beta)-d^2](a_o+u_o^{'}-c) - \\ (1-\beta)d(a_p+u_p^{'})\}^2 \end{pmatrix}}{8(1-\beta)^2(1-\beta-d^2)^2}. \quad \textbf{(24)}$$

As the expression for social welfare difference is very complex, this paper analyzes through a numerical example. Setting $a_p = a_o = 100$, $q_o^0 = 25$, $q_p^0 = 50$, $\alpha = \beta = 0.125$, $k_1 = k_2 = 0$ and $d = 0.5$, the two solutions of $(SW - SW')(c) = 0$ are given by $c_1^* = 27.6081$ and $c_2^* = 63.6014$. Figure 1 presents the social welfare difference as a function of the learning (maintenance or development) costs of open source software.

According to Figure 1, $SW > SW'$ when $c_1^* < c < c_2^*$ and $SW < SW'$ when $c < c_1^*$ or $c > c_2^*$. That is, there exists an interval of the learning (maintenance or development) costs of open source software, in which the social welfare is higher when proprietary software monopolizes the market than when it competes with open source software. Moreover, if the learning (maintenance or development) costs of open source software are low enough (or high enough), the social welfare is lower when proprietary software monopolizes the market than when it competes with open source software. Therefore, the emergence of open source software in a software market monopolized by proprietary software firm doesn't necessarily increase the social welfare, what happens in reality depends on the learning (maintenance or development) costs of open source software.

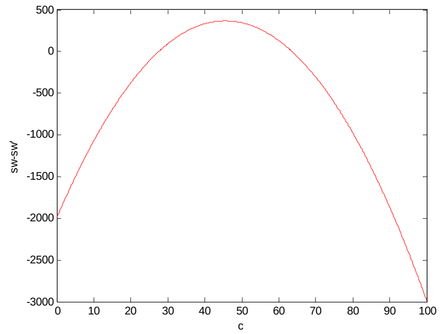

Figure 1

The impact of the learning (maintenance or development) costs of open source software on the social welfare difference.

CONCLUSIONS

To analyze how open source software impacts proprietary software firms' profit and social welfare, two models have been set up in this study, one of which is proprietary software monopolizing the software market and the other one of which is proprietary software competing with open source software. This paper assumes that proprietary software firm pursues profit maximization, open source software is free for users and the software market presents network externalities. Comparing the optimal results for two models, it mainly finds that: (i) the appearance of open source software in a software market does not necessarily decrease (resp. lower) the quantity and profit (resp. price) of proprietary software firm; (ii) if proprietary software isn't compatible to open source software, proprietary software firm prices higher, outputs more and obtains greater profit when proprietary software monopolizes the market than it competes with open source software; (iii) the appearance of open source software doesn't necessarily increase the social welfare.

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support from the Natural Science Foundation of Shandong Province (No. ZR2013GL005), the Social Science Planning Research Project of Shandong Province (No. 12CJRJ17) and the Shandong Higher School of Humanities and Social Science Research Projects (No. J13WF11).

REFERENCES

[1] Netcraft; September 2001 web server survey. Available: http://survey.netcraft.com/Survey/index-200109.html, **(2001)**.
[2] Netcraft; November 2006 web server survey. Available: http://news.netcraft.com/archives/web_server_survey.html, **(2006)**.
[3] S.Weber; The Success of Open Source. Harvard University Press, **(2004)**.
[4] T.O'Reilly; Lessons from Open-source Software Development. Communications of the ACM, **42(2)**, 33-37 **(1999)**.
[5] J.Dalle, N.Jullien; Open-source vs. Proprietary Software. Available: http://opensource.mit.edu/online_papers.php.
[6] Z.Meng, S.Y.Lee; Open Source vs. Proprietary Software: Competition and Compatibility. SSRN eLibrary. Available: http://ssrn.com/paper=780804, **(2005)**.
[7] M.Q.Xing; Game Analysis of Compatible Decisions between Open Source and Proprietary Software. Proceedings of the 2nd International Conference on Information Science and Engineering, **(2010)**.

[8] M.Mustonen; When Does a Firm Support Substitute Open Source Programming. Journal of Economics and Management Strategy, **14(1)**, 121-139 **(2005)**.
[9] L.H.Lin; Impact of User Skills and Network Effects on the Competition between Open Source and Proprietary Software. Electronic Commerce Research and Applications, **7(1)**, 68-81 **(2008)**.
[10] M.Q.Xing; Impact of Open Source Software on the Quality of Proprietary Software and Software Differentiation. Journal of Convergence Information Technology, **7(20)**, 242-249 **(2012)**.
[11] M.Katz, C.Shapiro; Network Externalities, Competition, and Compatibility. American Economic Review, **75(3)**, 424-440 **(1985)**.
[12] A.L.Bowley; The Mathematical Groundwork of Economics, Oxford: Oxford University Press, **(1924)**.
[13] P.Regibeau, K.E.Rockett; The Timing of Product Introduction and the Credibility of Compatibility Decisions. International Journal of Industrial Organization, **14(6)**, 801-823 **(1996)**.