UNIVERSITÄT BERN

Faculty of Economics and Social Sciences

Department of Economics

On Platforms, Incomplete Contracts, and Open Source Software

Andras Niedermayer

07-07

October 2007

DISCUSSION PAPERS

Schanzeneckstrasse 1
Postfach 8573
CH-3001 Bern, Switzerland
http://www.vwi.unibe.ch

On Platforms, Incomplete Contracts, and Open Source Software

Andras Niedermayer*

October 22, 2007

Abstract

We consider a firm A initially owning a software platform (e.g. operating system) and an application for this platform. The specific knowledge of another firm B is needed to make the platform successful by creating a further application. When B's application is completed, A has incentives to expropriate the rents. Netscape claimed e.g. that this was the case with its browser running on MS Windows. We will argue that open sourcing or standardizing the platform is a warranty for B against expropriation of rents. The different pieces of software are considered as assets in the sense of the property rights literature (see Hart and Moore (Journal of Political Economy, 1990)). Two cases of joint ownership are considered beyond the standard cases of integration and non-integration: platform standardization (both parties can veto changes) and open source (no veto rights). In line with the literature, the more important a party's specific investments the more rights it should have. In contrast to Hart and Moore, however, joint ownership can be optimal in our setting. Open source is optimal if investments in the applications are more important than in the platform. The results are driven by the fact that in our model firms invest in physical (and not in human) capital and that there is non-rivalry in consumption for software.

Keywords: Platforms, open source, standardization, incomplete contracts, property rights, joint ownership
JEL-Classification: C70, D23, L13, L22, L86

*Economics Department, University of Bern, Schanzeneckstrasse 1, CH-3001 Bern, Switzerland. Email: niedermayer@vwi.unibe.ch. I thank Philipp Ackermann, Marc Blatter, Kevin Boudreau, Winand Emons, Florian Englmaier, Claude Fluet, William Fuchs, Maija Halonen-Akatwijuka, Simon Loertscher, Gerd Muehlheusser, Alain Raybaut, and Andreas Roider for helpful comments. I further thank participants of the IIOC 2007 in Savannah, FLOSS Workshop 2007 in Nice, EEA/ESEM 2007 in Budapest, EARIE 2007 in Valencia, and seminars at the University of Bern for valuable discussions.

> "... If you look at history ... rumor has it that IBM spent a billion dollars on applications for [their operating system] OS2 but they bought people, they said 'here, do this for me.' And at the end, the companies turned around and said 'here, it's done' – and IBM said 'well, aren't you going to sell it, market it?' They said 'well no, our deal was to develop it: you've got it, now good luck.'"[1]

1 Introduction

Complementary products are crucial for the success of platforms in the IT industry.[2] Firms developing platforms typically provide some of the complementary products themselves or pay other firms to do so. However, as the quote at the beginning exemplifies, this is often not enough. Employees or firms that are simply paid to finish a product (e.g. a piece of software or a hardware peripheral[3]) do not have the same incentive to work hard for the success of the product as an owner of the product would. This is especially true for the IT industry where the complexity of the products and the unpredictability of the development processes make complete contingent contracts impossible. Therefore, the platform owner needs independent firms to create a complementary product.[4] But this raises problems of its own. After the independent firm has created the product and made the platform successful, the platform owner has the stronger position and *ex post* every incentive to expropriate the rents. An independent firm anticipating this may not be

[1] Claude Leglise, director of Intel Developer Relations Group, explaining why Intel wants independent firms to invest in products complementary to the Intel platform by themselves. Quote taken from Gawer and Henderson (2007, p. 20).

[2] The most prominent example of a platform is Microsoft Windows. We use platform in a similar sense as e.g. West (2003): standardized products that "allow modular substitution of complementary assets such as software and peripheral hardware". Corresponding examples of complementary products would be Microsoft Excel and Lotus 1-2-3 for Windows.

[3] A peripheral is a device attached to a computer such as a mouse or a printer.

[4] An economic explanation of why a platform owner wants other firms to enter its market and examples where this observation applies are given in Niedermayer (2006). Similar results are obtained in Economides (1996) for industries with network effects. Note that non-contractable effort level is not explicitly modeled in both papers, it is assumed that one cannot hire an employee to do the job.

willing to create the product ex ante.[5]

We will argue here that the different licensing schemes of platforms[6] and the different ownership structures[7] observed in the industry are commitment devices to reduce the inefficiencies created by this hold-up problem and to improve incentives to invest in innovation.

Many of the phenomena can be explained by the incomplete contracts and property rights literature à la Grossman and Hart (1986) and Hart and Moore (1990). If a firm's specific investment is more important, it should be assigned stronger property rights ex ante to have a stronger bargaining position ex post and hence higher investment incentives. However, some of the observations cannot be explained by the standard Hart and Moore framework. In the standard framework joint ownership and ownership by none can never be optimal.[8] We will argue that a standardized platform is analogous to joint ownership (all parties in the standardization committee have veto powers[9]) and that open source is analogous to ownership by none (no one has veto power). A further point where our analysis departs from the standard framework is the consideration of the asset creation phase. It is a recurring pattern that a start-up develops a first version or prototype of a product (e.g. piece of software) and then it is bought up by a large platform firm.

We explain these observed ownership structures with a model extending the Hart and Moore (1990) property rights framework. Hart and Moore have two managers of two firms (in our case one should rather think of

[5]This has also been called the "tension between appropriation and adoption" in the empirical literature on platforms, see e.g. West (2003).

[6]Proprietary, standardized, and open source; to be specified in detail later.

[7]Vertical integration, independent firms, initially independent startups being bought up later.

[8]Note that we consider property rights as rights to veto the usage of an asset similarly to Hart and Moore.

[9]Of course this is a strong simplification as other decision rules than unanimous vote can be agreed on for a committee. However, decisions made by a committee are in any case more complicated than those by a single party.

two software developers), say A and B.[10] At stage 1 ownership rights are decided on, then developers make specific sunk non-verifiable investments, and at stage 2 A and B renegotiate the distribution of profits. We introduce a further stage 0. At stage 0 A owns a monopoly platform and a complementary asset. We will refer to complementary assets as "applications" from now on.[11] Software developer A first decides whether the platform should be proprietary, standardized, or open source, which will have an impact on bargaining power later. B considers developing a first version or prototype of a new application. B can further decide whether he[12] wants to do this as an independent developer or whether he wants to negotiate with A about being hired. Stages 1 and 2 follow Hart and Moore (1990). At stage 1, if B is independent, A and B decide whether B should be acquired by A or stay independent.[13] After deciding on ownership structure, A and B make non-verifiable investments in the further development of the platform and the applications.[14] At stage 2 A and B renegotiate their contracts. Because we assume physical capital, an acquirer can make full use of the asset no matter whether the acquired agrees or not, and can therefore expropriate the whole rent.[15] If A's platform is proprietary and B is independent, A can threaten B to use her strong position to reduce B's application profits. Such an action has been referred to as an "ex post squeeze" of B's

[10] In the tradition of Hart and Moore (1990) we will abstract from many real world aspects of the relation of two firms. One abstraction is that there is only one manager in each firm – or that the management of a firm makes joint decisions that maximize the joint utility of this firm's management.

[11] This is for the sake of simplicity. Our model applies to more general complementary assets than just software applications. Such assets can also be a factory to create hardware or skills to provide consulting and support services complementary to the platform.

[12] In the following we will refer to A as "she" and to B as "he".

[13] We assume that A's specific investments are so important that it can never be efficient for B to acquire A.

[14] Imagine e.g. Internet compatibility coming up as a new requirement by users.

[15] Assuming physical capital strongly simplifies our analysis. However, similar results are likely to hold for human capital.

profits.[16,17] Microsoft was e.g. accused of not revealing parts of its new application programming interfaces (APIs) to competitors in the anti-trust case. If the platform is standardized[18] both A and B can threaten to veto mutually beneficial changes to the platform. If the platform is open source, no one has a threat. Stronger property rights result in a better bargaining position at renegotiations. The main statement of the model is that the more important B's specific investment the stronger his bargaining position should be. Bargaining power is the lowest if B has been acquired, higher if he is an independent developer for a proprietary platform, again higher for a standardized platform, and highest for an open source platform. In contrast to this, if the development of the platform or A's application is important, she should be given stronger bargaining power. We also get the following result. Assume specific investment in the first version of B's application (asset creation) is important, the further development, however, is not. Then it is optimal that B develops a first version as an independent firm and is acquired by A afterwards.

Related Literature. This paper is clearly related to the property rights and incomplete contracts literature (as in Grossman and Hart (1986), Hart and Moore (1990), and Hart (1995)). We have outlined above where we

[16] Gawer and Henderson (2007) describe in their case study that Intel managers considered it as crucial for the success of the Intel platform to "signal that [Intel] expects both Intel and its competitors to make money in complementary markets – that it will not 'suck all the air' out of them" (p. 3). "Squeezing" could be either decreasing the price in the application market (and thus increasing profits in the platform market) or "manipulating the boundary between the market and the core of the platform to advantage Intel-owned products" (p. 17). We will rather focus on the latter as this is more likely for the examples we have in mind: Apple, IBM, Sun (see West (2003)), and Microsoft.

[17] Farrell and Katz (2000) consider different ways of reducing a complementor's profits that they label price squeeze, investment squeeze, exclusionary squeeze, and extraction of side-payments (or access charges) by threat of a squeeze. As the exact way how the complementor's profits are reduced does not matter for our analysis, we refer the reader to Farrell and Katz (2000) for a detailed description.

[18] Examples include the Open Software Foundation and the X/Open consortium for the standardization of Unix operating system (see e.g. West (2003)) and the Java Community Process Executive Committee deciding about changes of the Java platform.

have departed from the standard framework. The main difference in results is that joint ownership and ownership by none can be optimal in our setup. Rosenkranz and Schmitz (1999) show that joint asset ownership with veto power can be optimal if know-how disclosure by both parties is necessary. One of the examples of ex post squeezing – not revealing the APIs – could be interpreted as non-disclosure of know-how. However, in our set-up there are three assets and only one party invests in platform development and can hence disclose its know-how.

Besides the incomplete contracts literature, this paper is also related to the growing literature on open source software, as surveyed and extended in Lerner and Tirole (2005) and Maurer and Scotchmer (2006). Our model has similarities with Bessen's (2005) who also considers open source software development in the context of incomplete contracts. However, the focus is on the complexity of the software and not on the ownership of the platform[19] and the applications running on it. Here we assume that the application sold by the platform owner is proprietary and only the platform is possibly open source. Polanski (2007, forthcoming) compares investment incentives in open source and proprietary projects in a framework of sequential bargaining whereas our focus is on the optimal allocation of property rights. Economides and Katsamakas (2006) look at open source programmers that maximize consumer surplus and their reputation and compare their investment incentives with those of developers of proprietary software.

Our paper also relates to the literature on complementary products. Farrell and Katz (2000) focus on the hold-up problem arising when a monopolist has an incentive to squeeze prices in a complementary market after the complementors made their investments.[20] Our focus is on how firms try to solve similar hold-up problems. The growing empirical literature on

[19]There is of course a connection between the two topics. Complexity is responsible for the fact that a platform cannot incorporate all the features required by users and, therefore, third-party applications running on it are necessary.

[20]See footnote 17 for other ways of reducing complementors' profits.

markets with complementary products supports our theoretical arguments in this paper. West (2003) contains case studies of three firms (IBM, Apple, Sun Microsystems) producing platforms and complementary products and describes their strategies to solve the hold-up problem[21] by open sourcing or standardizing their platforms. Gawer and Henderson (2007) describe Intel's strategy of encouraging entry to their platforms by credibly committing not to squeeze them ex post. They find that Intel's commitment devices are its internal organizational structure,[22] lowering the costs of entry largely by dissemination of intellectual property, and stability of subsidies to complementors. Boudreau's (2006) qualitative and quantitative analysis of mobile computing platforms[23] comes to empirical results that fit our theoretical predictions very well. Boudreau's main findings are that (1) platform suppliers change the boundaries of their platform[24] in order to induce entry by application developers and to give them investment incentives and that (2) a more open platform leads to higher investment in applications but lower investment in the platform itself.[25]

The paper is structured as follows. Section 2 introduces the basic setup. Section 3 describes ownership structures that can be agreed on at stage 1. Section 4 describes firms' investment choices between stages 1 and 2 and the properties of different ownership structures at stage 1. Section 5 describes stage 0 of the model when assets are created. Section 6 discusses the results

[21] Which West calls the "tension between appropriability and adoption".

[22] Intel clearly states that it distinguishes between units pursuing "Job 1", expanding demand for its microprocessor, and units pursuing "Job 2", making profits with markets complementary to its platform. It is also notable that Intel founded a stand alone, not-for-profit unit – the Intel Architectural Labs – which is responsible for the development of intellectual property benefiting the Intel platform and its complementors (Gawer and Henderson 2007, p.18).

[23] This includes handheld computers such as the Palm Pilot and smartphones that are able to access the Internet and load application software.

[24] Expanding the boundary of the platform is similar to hiring or acquiring an application developer in our model.

[25] Boudreau's results can be understood this way if one interprets "low coordination investment" (Boudreau 2006, p.90) as investment in applications and "high coordination investment" (p.92) as investment in the platform.

and Section 7 concludes.

2 Basic Setup

Our model is based on Hart (1995) and Hart and Moore (1990) where two managers of two firms (or software developers) owning two assets decide on asset ownership rights at stage 1, then make relation specific investments, and finally renegotiate their contract at stage 2. Here we deviate from Hart and Moore (1990) in three main points. First, we have a stage 0 when commitment to a software development model (open-source, standardized, non-standardized) is made and an asset is created. Second, there are three rather than two assets: besides the two applications (representing the assets of the two firms) there is also a platform whose development benefits both applications. Third, investment is in physical rather than human capital. This means that the owner of an asset can fully expropriate its rents and a non-owner has no interest in making any specific investments in the development of this asset.

The three assets are applications a and b, and a platform p as depicted in Figure 1. The two firms in the market are A's and B's firms. At stage 0 A has already developed application a and platform p, B considers developing application b. At stage 1, A and B decide on ownership structure of the assets and then on how much to invest in software development. At stage 2 there are renegotiations and demand and profits are realized. First, we will describe stages 1 and 2. It will be assumed that B did develop the application. Then we consider how expectations of what happens at stages 1 and 2 influence both developers' decisions at stage 0: A can choose a development model for her platform, B may or may not enter the market. Figure 2 depicts the timing.

Ownership rights mean in our setting that one has a residual veto power. The exact meaning of veto power will be specified later. In the following

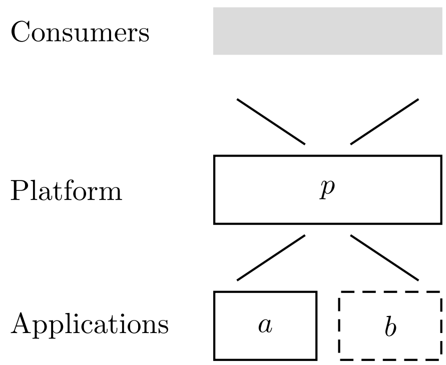

Figure 1: Applications a and b; platform p. a and p are initially owned by firm A, b is initially owned by firm B. Firm B may or may not enter the market.

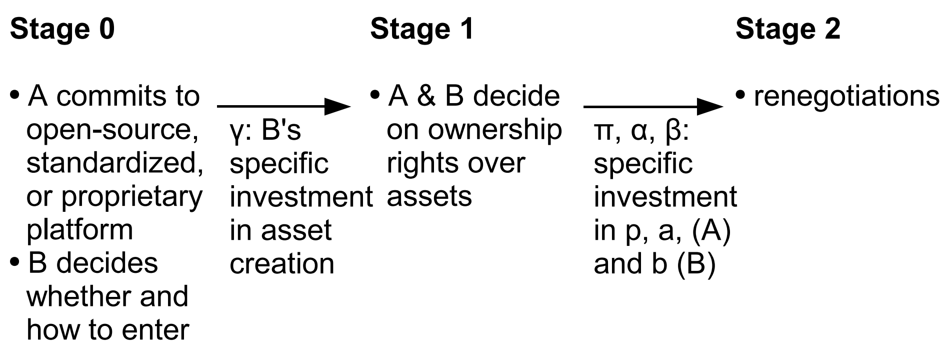

Figure 2: Timing of the model

"$\mathbb{X} = \{y, z\}$" will mean "developer X has veto power over assets y and z". Following four types of ownership structure are possible at stage 1.

- Acquisition: A acquires B[26] and owns both applications and the platform ($Q := (\mathbb{A} = \{a, b, p\}, \mathbb{B} = \emptyset)$)
- Proprietary Platform with Independent Application Developer: A owns the platform and application a, B owns application b ($P := (\mathbb{A} = \{a, p\}, \mathbb{B} = \{b\})$)
- Standardized Platform with Independent Application Developer: A

[26] For the sake of simplicity we will say "A acquires B" throughout the paper. To be precise it should of course read "A's firm acquires B's firm".

owns application a, B owns application b, joint ownership of platform, both firms have veto rights ($S := (\mathbb{A} = \{a, p\}, \mathbb{B} = \{b, p\})$)

- Open Source Platform with Independent Application Developer: A owns application a, B owns application b, joint ownership of platform, none has veto power over platform ($O := (\mathbb{A} = \{a\}, \mathbb{B} = \{b\})$)

There are further possibilities of ownership structure which will not be considered here.[27]

Between stages 1 and 2 A and B invest in the development of the applications and the platform. An application benefits from an improvement of the platform. At stage 2 firms renegotiate contracts. If they reach an agreement, changes to the platform are implemented and both firms benefit from the improvement. If negotiations break down, the owner of the platform can reduce the revenues of the application b, by using an "ex post squeeze" as described in Farrell and Katz (2000) or by denying access to the new Application Programming Interface (API) of the platform.[28] In case of a standardized platform changes to the platform can be blocked and both firms benefit less from improvements.[29]

Revenues generated by the applications are determined by investments in the development of the platform p and applications a and b, denoted as π, α, $\beta \in \mathbb{R}_0^+$, respectively. We will assume that developers make non-verifiable investments in physical capital which can thus be fully expropriated by the asset owner.[30] A has the specific knowledge to make

[27] E.g. B may acquire A or one might give ownership of the platform to an independent firm. We will assume that the specific investment of A in her application and the platform are so important that these ownership structures can never be optimal and hence need not be considered. Note further that whether the platform is proprietary, standardized, or open source only matters when there is an independent developer for this platform.

[28] An API is the interface through which an application developer accesses the platform. Having access to the old API only gives an application developer a competitive advantage.

[29] Integrating modules for a simplified access to the Internet into the platform may be a feature desired by consumers. If such features are not incorporated, demand for the platform and hence both applications may decrease.

[30] Physical capital implies hence that the investor does not gain bargaining power by

investments α and π, B has the specific knowledge for investment β. The only purpose of the platform is to increase applications' revenues, it does not create revenues itself.[31] We further assume that both applications' revenues depend on the investment in the platform. We will denote revenues created by applications a and b in case of an agreement as $R_a(\alpha, \pi)$ and $R_b(\beta, \pi)$. To simplify notation we will write first and second derivatives as $R_{kl}(k', \pi) := \partial R_k(k', \pi)/\partial l$ and $R_{klm}(k', \pi) := \partial R_k(k', \pi)/\partial m \partial l$ for $(k, k') \in \{(a, \alpha), (b, \beta)\}$ and $l, m \in \{k', \pi\}$. We assume revenue functions to be twice differentiable, increasing ($R_{a\alpha} > 0$, $R_{a\pi} > 0$, $R_{b\beta} > 0$, $R_{b\pi} > 0$) and concave ($R_{a\alpha\alpha} < 0$, $R_{a\pi\pi} < 0$, $R_{b\beta\beta} < 0$, $R_{b\pi\pi} < 0$) in investments. Further, we assume that an investment does not affect an other investment's marginal revenue ($R_{a\alpha\pi} = R_{b\beta\pi} = 0$).[32] Because cross-derivatives are zero we will drop the unnecessary variable when writing the first derivative, e.g. $R_{a\pi}(\pi) := R_{a\pi}(\alpha, \pi)$.

In case of a break down of negotiations generated revenues are $r_a^{(\mathbb{A},\mathbb{B})}(\alpha, \pi)$ and $r_b^{(\mathbb{A},\mathbb{B})}(\beta, \pi)$, where $(\mathbb{A}, \mathbb{B})$ describes the ownership structure in the industry. The same notation for the derivatives is used as for R_a and R_b and we assume $r_{a\alpha}, r_{a\pi}, r_{b\beta}, r_{b\pi} \geq 0$, $r_{a\alpha\alpha}, r_{a\pi\pi}, r_{b\beta\beta}, r_{b\pi\pi} \leq 0$, and $r_{a\alpha\pi} = r_{b\beta\pi} = 0$ for all ownership structures. If negotiations break down joint revenues are always less or equal to the case of successful negotiations:

$$R_a + R_b \geq r_a^{(\mathbb{A},\mathbb{B})} + r_b^{(\mathbb{A},\mathbb{B})}, \qquad \forall \alpha, \beta, \pi, \mathbb{A}, \mathbb{B}.$$

investing because the owner of the asset can use it without the investor's consent. One can extend the model to accommodate for human capital. In this case the investor can threaten to stop cooperating and human capital would be lost.

[31] This assumption is valid for Adobe as firm A, the (free) PDF format as the platform, and Adobe Acrobat Standard as the application (see the appendix in Niedermayer (2006)). It is not an unreasonable approximation for MS Windows, as the price of the applications sold by Microsoft are much higher than the price of the operating system.

[32] We also need the technical assumptions $R_{kl}(0) \geq 1$ and $R_{kl}(\infty) < 1$ for all k, l.

3 Ownership Structure

In the following we will describe the results of stage 2 renegotiations for the four ownership structures (1) acquisition of B by A, independent application development for (2) a proprietary, (3) a standardized, and (4) an open source platform. To simplify exposition, we will call them the acquisition, the proprietary platform, the standardized platform, and the open source platform cases. We will assume in all cases that if a surplus is generated by an agreement, it will be divided according to the Nash bargaining solution so that both A and B get half of the surplus. We will consider ownership structures as though firms would choose between all four options at stage 1. However, as we will see later, if standardization or open sourcing makes sense, A commits to it already at stage 0. Therefore, there are actually only two possibilities to choose from at stage 1: acquisition or independent application development with a predetermined licensing of the platform.

Acquisition of B by A. Because investment is in physical capital, it can be fully expropriated by the asset owner. Therefore, B has no threat in the bargaining process and revenues in case of a break down of negotiations are

$$r_a^Q = R_a + R_b,$$
$$r_b^Q = 0.$$

As A has all the bargaining power, she will extract the whole surplus. Ex post profits of A and B are hence

$$V_A^Q = R_a + R_b,$$
$$V_B^Q = 0.$$

Proprietary Platform with Independent Application Developer. In case negotiations break down, the same profit can be achieved with application a. B, however, has lower revenues because of the "ex post squeeze"

A engages in.

$$r_a^P = R_a,$$
$$r_b^P = (1-\kappa)R_b, \qquad \kappa \in (0,1).$$

The surplus created by an agreement, κR_b, is split according to the Nash bargaining solution, thus ex post profits are

$$V_A^P = R_a + \frac{\kappa}{2}R_b,$$
$$V_B^P = \left(1-\frac{\kappa}{2}\right)R_b.$$

Standardized Platform with Independent Application Developer
If negotiations break down, none of the firms can use the new API of the platform. Revenues are

$$r_a^S = (1-\lambda)R_a,$$
$$r_b^S = (1-\mu)R_b,$$

with $\lambda, \mu \in (0,1)$.

The surplus is again split evenly and ex post profits are

$$V_A^S = \frac{1}{2}(R_a - r_a + R_b - r_b) + r_a = \left(1-\frac{\lambda}{2}\right)R_a + \frac{\mu}{2}R_b,$$
$$V_B^S = \frac{1}{2}(R_a - r_a + R_b - r_b) + r_b = \frac{\lambda}{2}R_a + \left(1-\frac{\mu}{2}\right)R_b.$$

Open Source Platform with Independent Application Developer
Here no one has a veto right, therefore, it is irrelevant whether an agreement is reached and

$$r_a^O = R_a,$$
$$r_b^O = R_b.$$

Ex post profits are accordingly

$$V_A^O = R_a,$$
$$V_B^O = R_b.$$

4 Investment Choice

We will first describe first-best levels of investment and then actual investment for the different ownership structures.

First-best. In first-best joint ex ante profits would be maximized

$$\max_{\pi,\alpha,\beta} R_a + R_b - \pi - \alpha - \beta.$$

The first-order conditions

$$\begin{aligned} R_{a\pi}(\pi^*) + R_{b\pi}(\pi^*) &= 1, \\ R_{a\alpha}(\alpha^*) &= 1, \\ R_{b\beta}(\beta^*) &= 1, \end{aligned} \tag{4.1}$$

are sufficient to determine the optimum because of the concavity of the revenue functions.

In a second-best world, A and B invest such that their investments maximize ex ante profits

$$\begin{aligned} &\max_{\pi,\alpha} \quad V_A^{(\mathbb{A},\mathbb{B})} - \alpha - \pi, \\ &\max_{\beta} \quad V_B^{(\mathbb{A},\mathbb{B})} - \beta. \end{aligned}$$

We get the following equilibrium conditions by setting the derivatives of the ex post profits from Section 3 minus investment costs to zero, except for the corner solution β^Q in the acquisition case.

Acquisition

$$\begin{aligned} R_{a\pi}(\pi^Q) + R_{b\pi}(\pi^Q) &= 1, \\ R_{a\alpha}(\alpha^Q) &= 1, \\ \beta^Q &= 0. \end{aligned} \tag{4.2}$$

Proprietary Platform

$$R_{a\pi}(\pi^P) + \frac{\kappa}{2}R_{b\pi}(\pi^P) = 1, \tag{4.3}$$
$$R_{a\alpha}(\alpha^P) = 1,$$
$$\left(1 - \frac{\kappa}{2}\right) R_{b\beta}(\beta^P) = 1.$$

Standardized Platform

$$\left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi^S) + \frac{\mu}{2}R_{b\pi}(\pi^S) = 1, \tag{4.4}$$
$$\left(1 - \frac{\lambda}{2}\right) R_{a\alpha}(\alpha^S) = 1,$$
$$\left(1 - \frac{\mu}{2}\right) R_{b\beta}(\beta^S) = 1.$$

Open Source Platform

$$R_{a\pi}(\pi^O) = 1, \tag{4.5}$$
$$R_{a\alpha}(\alpha^O) = 1,$$
$$R_{b\beta}(\beta^O) = 1,$$

4.1 Comparison of Investment Levels

In the following we will compare investment levels for the different ownership structures. For the sake of simplicity we only consider the cases where investments have the same influence on both application revenues, i.e. $R_{a\pi}(\alpha) = R_{b\pi}(\beta)$. It is useful to consider first the case with $\lambda < \mu < \kappa < 1$. This is the situation where in case of no agreement firms still benefit from improvements of the platform, the effect of no agreement is less severe than in the case of an independent application developer, and the developer of the platform loses less in case of a break down of negotiations. The advantage of considering this situation is that any of the ownership structures can be optimal depending on the importance of these different investments α, β, and π. We will show later that for other cases some ownership structures are dominated.

Proposition 1 states that investments are (weakly) less than first-best investments and gives a ranking of investment levels for different ownership structures. Table 1 summarizes the results.

Proposition 1. *Investments for different ownership structures compare as follows:*

i) $\pi^* = \pi^Q \geq \pi^P \geq \pi^S \geq \pi^O$,

ii) $\alpha^* = \alpha^Q = \alpha^P = \alpha^O \geq \alpha^S$,

iii) $\beta^* = \beta^O \geq \beta^S \geq \beta^P \geq \beta^Q$.

Proof. The proof follows Hart (1995, p. 41). Note that inequalities are weak to accommodate for the case that optimal investment is zero.

i) For any π we have for the left-hand-sides of Eqs. (4.1), (4.2), (4.3), (4.4), and (4.5)

$$\begin{aligned} &R_{a\pi}(\pi) + R_{b\pi}(\pi) > R_{a\pi}(\pi) + \frac{\kappa}{2} R_{b\pi}(\pi) \\ &> \left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi) + \frac{\mu}{2} R_{b\pi}(\pi) > R_{a\pi}(\pi). \end{aligned} \tag{4.6}$$

From (4.1), (4.2), (4.3), (4.4), and (4.5) we also know that in all equations the left-hand-sides are equal to 1 in optimum, therefore,

$$\begin{aligned} &R_{a\pi}(\pi^*) + R_{b\pi}(\pi^*) = R_{a\pi}(\pi^Q) + R_{b\pi}(\pi^Q) \\ &= R_{a\pi}(\pi^P) + \frac{\kappa}{2} R_{b\pi}(\pi^P) = \left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi^S) + \frac{\mu}{2} R_{b\pi}(\pi^S) = R_{a\pi}(\pi^O) \end{aligned} \tag{4.7}$$

We can use the fact that for concave functions $f(\cdot)$ and $g(\cdot)$ $(f'(x) > g'(x), \forall x$ and $f'(x_1) = g'(x_2))$ implies $(x_1 > x_2)$. The proposition follows from (4.6) and (4.7) and the concavity of R_a and R_b in π ($R_{a\pi\pi} < 0$, $R_{b\pi\pi} < 0$).

ii) Analogously.

iii) Analogously. Additionally, $\beta^Q \leq \beta^P$ follows from $\beta^Q = 0$ and $\beta \geq 0$. □

Proposition 1 means that A's incentive to invest in platform development is highest if it has acquired B, lower if B is independent and the platform

	π	α	β
First-best (*)	1	1	1
Acquisition (Q)	1	1	4
Proprietary Platform (P)	2	1	3
Standardized Platform (S)	3	2	2
Open Source Platform (O)	4	1	1

Table 1: Ranking of investment levels for the platform and applications a and b depending on ownership structure. A lower number means higher investment.

proprietary, even lower for a standardized, and lowest for an open source platform. B's incentives to invest in his application are ranked in the opposite order. In our setup, A's investment in her own application is first-best except for standardization, where A has to fear that some of the rents generated by the application might be expropriated by B. The result can be interpreted as stating that the firms should be integrated if investment in platform quality is the most important issue. If investment in B's application is most important, the platform should be open source. For intermediate cases, A and B will agree on a proprietary or a standardized platform ex ante.

This seems to fit some observations in the industry. If one considers e.g. user friendliness as a task that has to be achieved by investments in the platform then one would expect the following. A tightly integrated platform such as Apple's Mac OS should be user friendly, but have rather low investments in third-party applications. A proprietary platform such as Microsoft Windows would be somewhat less user friendly and have higher investments by independent application developers. A standardized platform such as Sun's Java Platform and an open source platform such as the Apache web server[33] would be less user friendly, but would have relatively larger investments in applications by independent developers.[34]

[33] A web server allows users to access web pages. However, for many web servers add-on modules with further functionality are available such as database access.

[34] It has to be noted that investments in independent applications for Microsoft Windows

4.2 Comparisons for Other Parameter Values

In the following we will keep the assumption $R_{a\pi} = R_{b\pi}$ and consider investment levels for different values of κ, λ, and μ. It is interesting that for some of these cases, standardization is dominated.

$\mu < \lambda$: (*A blocking of changes in case of standardization does more harm to A than to B.*) Because of $(1 - \lambda/2)R_{a\pi} + (\mu/2)R_{b\pi} < R_{a\pi}$ we have $\pi^S < \pi^O$ by the same argument as in the proof of Proposition 1. As the order of the other investments remains the same, for a standardized platform all investment levels are lower than for open source. Changing the order of π^S and π^O in Table 1 reveals that standardization is dominated by other ownership structures in this case and thus cannot be optimal no matter which investment is more important.

$\mu \in (\kappa, \kappa + \lambda)$: (*A blocking of changes in case of standardization causes more harm to B than an ex post squeeze in case of a proprietary platform. But the difference between the blocking harm for B and for A is less than B's harm in case of an ex post squeeze.*) Because of $(1-\kappa/2)R_{b\beta} > (1-\mu/2)R_{b\beta}$ investment β is less for a standardized than for a proprietary platform, while the order of the other investments remains the same. Thus standardization is dominated by a proprietary platform.

$\mu > \kappa + \lambda$: (*The difference between the blocking harm for B and for A is larger than the ex post squeeze harm.*) $(1 + \kappa/2)R_{a\pi} < (1 - \lambda/2 + \mu/2)R_{a\pi}$ implies $\pi^S > \pi^P$. $\beta^P > \beta^S$ from the previous paragraph still holds. This means that the ordering of a standardized and a proprietary platform with respect to both investments in the platform and the application b is reversed. This is because for this parameter combination A's bargaining power in case of standardization is larger than in case of a proprietary platform.

---

are larger on an absolute scale. However, the Windows market is also much larger than the markets of the two last mentioned platforms, so are Microsoft's investments in the platform and in its own applications. The large variety of complementary modules and extensions for the Apache web server is notable given the size of the web server market.

5 Asset Creation at Stage 0

So far, we have considered application b as already existent. Now we will consider its creation. How the first version of a piece of software is created is very important for its later success. We will assume that at stage 0 B makes the specific investment γ in the creation of application b. When incorporating the specific investment in asset creation in the revenue function it becomes $R_b(\beta, \gamma, \pi)$. Similar assumptions apply as before: $R_{b\gamma} > 0$, $R_{b\gamma\gamma} < 0$, $R_{b\beta\gamma} = R_{b\pi\gamma} = 0$. R_a is not affected by γ.

5.1 Expectations of Stage 1 Negotiations

To understand the decision about asset creation we need to look at acquisition negotiations at stage 1. Let us assume that if there is no acquisition at stage 1 either of the following three situations occurs: 1. A made no commitment at stage 0, B is an independent developer for the proprietary platform (P), 2. A standardized the platform at stage 0, B develops an application for the standardized platform (S), 3. A open sourced the platform at stage 0 (O). We will denote the no acquisition case with N, N being P, S, or O depending on A's previous commitment.

We will denote ex ante profits at stage 1 with

$$U_A^k := V_A^k - \alpha^k - \pi^k,$$
$$U_B^k := V_B^k - \beta^k,$$

with $k \in \{Q, N\}$. Assuming that negotiations are efficient, acquisition will happen if $U_A^Q + U_B^Q > U_A^N + U_B^N$. In case of an acquisition the generated surplus is divided according to the Nash bargaining solution, A and B each get $(U_A^Q + U_B^Q - U_A^N - U_B^N)/2$, additionally to the profit they would get without an acquisition. Therefore, A has the profit

$$U_A^N + \frac{U_A^Q + U_B^Q - U_A^N - U_B^N}{2} = \frac{U_A^Q + U_A^N + U_B^Q - U_B^N}{2}$$

and B's firm is bought for the price

$$U_B^N + \frac{U_A^Q + U_B^Q - U_A^N - U_B^N}{2} = \frac{U_B^Q + U_B^N + U_A^Q - U_A^N}{2}.$$

5.2 Stage 0

At stage 0 A first commits to either nothing ($N = P$), to a standardization of her platform ($N = S$), or to open sourcing her platform ($N = O$). Next B has to decide whether he wants to develop a first version of his application and if he does want to, whether he wants to be independent or bargain with A about being hired. If B does not enter the market he earns his outside option $\tilde{U}_B$; further, A's platform will be less successful as it has only one application and A's revenues will be $\tilde{U}_A$. If B chooses to develop his application, he makes a non-verifiable investment in physical capital γ. If B is an employee, he will have no bargaining power at stage 1 whatsoever, therefore, he will invest nothing. If B is independent and anticipates no acquisition at stage 1, he will invest such that $U_B^N - \gamma$ is maximized. His investment is hence given implicitly by $U_{B\gamma}^N = 1$. If he expects to be acquired at stage 1, he invests such that $(U_B^Q + U_B^N + U_A^Q - U_A^N)/2 - \gamma$ is maximized. It can be shown that if $R_{b\beta\gamma} = 0$ the optimal investment level is the same as when no acquisition is expected and γ is hence given again implicitly by $U_{B\gamma}^N = 1$.[35] If γ and β were complements (i.e. $R_{b\beta\gamma} > 0$), γ would be lower if B expects an acquisition.

Two observations can be made. First, as mentioned in Hart (1995), if the asset is owned by someone else than the creator of the asset, there would be too little effort invested in asset creation because of the hold-up at completion. Second, if N is chosen such that B as an independent firm has a better bargaining position at stage 1, B will invest more in the asset creation no matter whether there will be an acquisition at stage 1 or not.[36] This

[35] E.g. for $N = P$ the derivative of $(U_B^Q + U_B^N + U_A^Q - U_A^N)/2$ with respect to γ is $\partial[(0 + (1 - \kappa/2)R_b + R_b - (\kappa/2)R_b)/2]/\partial\gamma$ which is equal to $U_{B\gamma}^P$.

[36] And if we modify the model such that U_A increases in γ, this can be beneficial for A

shows us that if standardization or open sourcing (two measures increasing B's bargaining position at stage 1) makes sense at stage 1, it makes even more sense at stage 0, as it will increase B's incentives for an investment in asset creation.

As for the different investment levels at the asset creation stage, the same applies as for β: $\gamma^* = \gamma^O \geq \gamma^S \geq \gamma^P \geq \gamma^Q$, with the proof being completely analogous. Given the assumption $R_{b\beta\gamma} = 0$, investments are the same if an acquisition is expected at stage 1: $\gamma^{NQ} = \gamma^N$, $N \in \{P, S, O\}$.

6 Discussion

We sum up the different scenarios which are possible given the results described above.

B as employee: A hires B to develop application b at stage 0. B makes no non-verifiable investments in the development of the first version at stage 0 ($\gamma = 0$) and in the further development of the application at stage 1 ($\beta = 0$). A makes profits $U_A^Q(\pi = \pi^Q, \alpha = \alpha^Q, \beta = 0, \gamma = 0)$ and pays B his outside option at stage 0 in order to hire him. If it is not attractive for B to enter on his own ($U_B^P - \gamma^P < \tilde{U}_B$) the outside option is $\tilde{U}_B$.

B acquired at stage 1: B develops a first version of his application and is then acquired by A. B starts developing his application with the intention of being acquired later. Depending on the licensing of the platform (proprietary, standardized, or open source) he anticipates his bargaining position at acquisition and chooses his investment in the first version accordingly (γ^{NQ}, $N \in \{P, S, O\}$). After being acquired, B invests nothing in the further development of the application ($\beta = 0$). A makes profits $U_A^Q(\pi = \pi^Q, \alpha = \alpha^Q, \beta = 0, \gamma = \gamma^{NQ})$ and pays

even if there is no acquisition at stage 1.

$(U_B^Q + U_B^N + U_A^Q - U_A^N)/2$ from these for the acquisition of B at stage 1. The three possible licensing schemes for the platform are:

Proprietary. A does not make any commitments. B knows that the acquisition price will be lower and invests little (γ^{PQ}) in asset creation.

Standardized. If higher investment in asset creation is needed, A commits to standardize her platform, giving B hence a better bargaining position and higher investment incentives. Standardization may further serve to make entry more attractive for B if B's profits resulting from an application for a proprietary platform are not sufficient.

Open Source. If γ should be increased further or B's entry should be encouraged more, A commits to open source.

B stays independent: B creates his application as an independent developer and stays independent at stage 1. B anticipates his bargaining position at stage 2 already at stage 0 and makes his investment in the first γ^N and the newer β^N version of his application according to the licensing of the platform ($N \in \{P, S, O\}$). Investment β is clearly higher than in case of an acquisition. A's profits are $U_A^N(\pi = \pi^N, \alpha = \alpha^N)$ and B's are $U_B^N(\pi = \pi^N, \beta = \beta^N, \gamma = \gamma^N)$. As γ has little impact on A's profits in case of standardization and no impact in case of open source, one would expect A not to be willing to move from proprietary licensing if she does not intend to acquire B at stage 1. However, this is not the case: B might not be willing to enter the market if he can only achieve the low profits that development for a proprietary platform offers. Left with only one application running on it, A's platform will not be as successful, A earning thus only $\tilde{U}_A$. Therefore, all of the following three licensing schemes for the platform are possible.

Proprietary. A has high, B has low profits. If B's profits are too low ($U_B^P - \gamma^P < \tilde{U}_B$) he will opt for his outside option.

Standardized. A's profits are lowered, B's increased. If standardization lets B's entry become profitable ($U_B^P - \gamma^P < \tilde{U}_B < U_B^S - \gamma^S$) and entry is beneficial to A, A will standardize. This result only holds if side payments to independent firms in exchange for market entry are not feasible. Otherwise A will pay B just enough to be indifferent between entry and non-entry.

Open Source. If not even standardization is sufficient to induce entry, open sourcing may be (if $U_B^S - \gamma^S < \tilde{U}_B < U_B^O - \gamma^O$). Again, A would prefer side payments if possible.

Single Application Platform. It can occur that convincing B to enter would be so costly for A that he prefers non-entry. In this case A's platform is less successful and A only earns $\tilde{U}_A$, B earns his outside option $\tilde{U}_B$. Note that entry may not occur in second-best even if it is optimal in first-best. This is the case if $U_A^* + U_B^* - \gamma^* > \tilde{U}_A + \tilde{U}_B > U_A^X + U_B^X - \gamma^X$, X being any of the licensing/ownership structures mentioned above.

7 Conclusions

We have considered a model with a developer A whose firm owns both a platform and an application running on the platform. It would be important for the success of the platform if another developer B's firm created a further application for it. However, B may hesitate to do so, as the rents generated might be expropriated by A. If B enters, he may make suboptimal specific investments both when creating and when developing further his application because of the hold-up problem.

Depending on the importance of firms' specific investments at different stages, one of several possible outcomes will occur. A may hire B to create the application; B may stay independent at asset creation and be acquired by A later on; or B may stay independent at all stages. In both cases of B's independence, three licensing schemes can be chosen by A for her platform: proprietary, standardized, or open source. Standardization increases B's bargaining position, open sourcing increases it even further. The higher B's bargaining power, the more willing he is to enter and the more specific investments he will make in case of entry. At the same time, this reduces A's share of overall profits and investment incentives. As a further effect, A's profits may be reduced so far that she is not willing to make concessions to induce B's entry. In this case B will stay out of the market and A's application will be the only one to run on her platform. This can happen even if entry were optimal in first-best.

Facing the abovementioned alternatives, firms choose the ownership structure that minimizes inefficiency.

References

Bessen, J. E. (2005): "Open Source Software: Free Provision Of Complex Public Goods," *SSRN eLibrary.*

Boudreau, K. (2006): "How Open Should an Open System Be?: Essays on Mobile Computing," Ph.D. thesis, Massachusetts Institute of Technology, Sloan School of Management.

Economides, N. (1996): "Network externalities, complementarities, and invitations to enter," *European Journal of Political Economy*, 12(2), 211–233.

Economides, N., and E. Katsamakas (2006): "Linux vs. Windows: A Comparison of Application and Platform Innovations Incentives for Open Source and Proprietary Software Platforms," in *The Economics of Open Source Software Development: Analyzing Motivation, Organization, Innovation and Competitions in the Open Source Software Revolution*, ed. by J. Bitzer, and P. Schröder. Elsevier.

Farrell, J., and M. Katz (2000): "Innovation, Rent Extraction, and Integration in Systems Markets," *Journal of Industrial Economics*, 48(4), 413–432.

Gawer, A., and R. Henderson (2007): "Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel," *Journal of Economics & Management Strategy*, 16(1), 1–34.

Grossman, S., and O. Hart (1986): "The Costs and Benefits of Ownership: A Theory of Vertical and Lateral Integration," *The Journal of Political Economy*, 94(4), 691–719.

Hart, O. (1995): *Firms, Contracts, and Financial Structure*. Clarendon Press, Oxford.

Hart, O., and J. Moore (1990): "Property rights and the nature of the firm," *Journal of Political Economy*, (98), 1119–1158.

Lerner, J., and J. Tirole (2005): "The Economics of Technology Sharing: Open Source and Beyond," *Journal of Economic Perspectives*, 19(2), 99–120, available at http://ideas.repec.org/a/aea/jecper/v19y2005i2p99-120.html.

Maurer, S., and S. Scotchmer (2006): "Open Source Software: The New IP Paradigm," in *Handbook of Economics and Information Systems*, ed. by T. Hendershott, pp. 285–319. Elsevier.

Niedermayer, A. (2006): "Does a Platform Monopolist Want Competition?," Diskussionsschriften dp0604, Universitat Bern, Volkswirtschaftliches Institut, available at http://ideas.repec.org/p/ube/dpvwib/dp0604.html.

Polanski, A. (2007, forthcoming): "Is the General Public License a Rational Choice?," *Journal of Industrial Economics*.

Rosenkranz, S., and P. W. Schmitz (1999): "Know-how disclosure and incomplete contracts," *Economic Letters*, (63), 181–185.

West, J. (2003): "How open is open enough? Melding proprietary and open source platform strategies," *Research Policy*, 32(7), 1259–1285.