On Platforms, Incomplete Contracts, and Open Source Software

Andras Niedermayer*

December 20, 2006
revised April 4, 2007

PRELIMINARY AND INCOMPLETE.

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 Winand Emons, Simon Loertscher, and Gerd Muehlheusser for helpful comments.
1 Introduction

Consider a software developing firm – e.g. Microsoft – producing both a platform (MS Windows) and an application running on it (MS Excel). Niedermayer (2006) argues that Microsoft benefits if a further firm starts producing an application for Windows (e.g. Lotus 1-2-3), because it will increase sales of Windows and possibly even of Excel. This can be true even if Excel and Lotus 1-2-3 are competing products.

Such a situation leads to an interesting hold-up problem. Microsoft is interested in inviting a competitor to make the market for Windows applications larger. However, once the market is large, Microsoft wants to get a share as large as possible of this market. As Microsoft has the ownership rights over Windows it can alter the compatibility of the platform with the applications in its own favor or refuse revealing the Application Programming Interface (API) of new versions of Windows to competitors. The independent application developer anticipates this and is possibly unwilling to enter the market without further warranties by Microsoft. Such warranties may be an acquisition by Microsoft, standardization of the platform, or open sourcing the platform. However, these remedies have costs of their own. An acquisition reduces the formerly independent firm’s incentives to invest further in its application. Standardization of the platform (here: giving the independent firm the right to veto changes) leads to a new hold-up problem, causing underinvestment in the platform and the incumbent’s application. Open sourcing the platform (none of the firms has a right to veto changes) leads to a public good problem and underinvestment in platform quality.

1 An economic explanation and further examples where this observation applies are given in Niedermayer (2006). Similar results are obtained in Economides (1996) for industries with network effects. Of course, the presence of a further application is even more beneficiary if the applications are complements.

2 Roughly speaking, an API gives a software developer the possibility to connect his software to the functionality of another piece of software.
This article explains why we observe these different ownership structures with a model based on property rights à la Hart and Moore (1990). One of the main differences to Hart and Moore (1990) is that we have a stage 0 additionally to stages 1 and 2. At stage 0 firm A has already created a platform and an application for it. A first decides whether the platform should be proprietary, standardized, or open source, which will have an impact on bargaining power later. The owner of firm B considers whether he should create a first version or prototype of a new application (i.e. his asset) for this platform and if he does create it, under which institutional arrangements (as A’s employee or as an independent firm). Stages 1 and 2 follow Hart and Moore (1990). At stage 1, if B is still independent, A and B decide whether B should be acquired by A or stay independent. After deciding on ownership structure, firms make non-verifiable investments in the further development of the platform and the applications. At stage 2 firms 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, therefore, the acquired firm has no threat. If A’s platform is proprietary and B is independent, he can threaten to deny B access to his platform’s API; if the platform is standardized, both firms can threaten to veto changes to the platform; if the platform is open source, no firm has any threat. Firms with more property rights have a better bargaining position at the 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. Conversely, if the development of the platform or A’s application is important, he should be given stronger bargaining power. A further result is that it can be optimal  

\[3\] This is a further difference to Hart and Moore (1990), instead of two, we have three assets: A’s platform and application, and B’s application.
that B develops a first version and then is acquired by the platform owner if specific investment in the first version of B’s application (asset creation) is important, the further development, however, is not.

**Related Literature.** This paper will argue in line with the property rights literature (as in Hart (1995) and Hart and Moore (1990)) that ownership rights will be allocated such that ex ante inefficiency is minimized. 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 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. Rosenkranz and Schmitz (1999) show that joint asset ownership with veto power can be optimal if know-how disclosure by both parties is necessary. Revealing the API of the platform is similar to know-how disclosure, however, in our set-up there are three assets and only one party invests in platform development and can hence disclose its know-how.

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 and Section 7 concludes.

## 2 Basic Setup

Our model is based on Hart (1995) and Hart and Moore (1990) where two firms owning two assets decide on asset ownership rights at stage 1, then

---

4There 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.
make relation specific investments, and finally renegotiate their contract at stage 2. The main result of Hart and Moore (1990) is that the more important a firm’s specific investments the stronger its bargaining position at stage 2 (and hence the more extensive its property rights) should be. 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 and B. At stage 0 firm A has already developed application \( a \) and platform \( p \), B considers developing application \( b \). At stage 1, firms 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 firms’ decisions at stage 0: A can choose a development model for its 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 \( "X = \{y, z\}" \) will mean “firm X has veto power over assets y and z”. Following four types of ownership structure are possible at stage 1.

- **Acquisition:** A acquires B and owns both applications and the platform 

\[
(Q := (A = \{a, b, p\}, B = \emptyset))
\]
Consumers

Platform p

Applications a and b

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.

Stage 0
- A commits to open-source, standardized, or proprietary platform
- B decides whether and how to enter

Stage 1
- A & B decide on ownership rights over specific assets
- π, α, β: specific investment in p, a, (A) and b (B)

Stage 2
- Renegotiations

Figure 2: Timing of the model

- Proprietary Platform with Independent Application Developer: A owns the platform and application a, B owns application b (P := (A = \{a, p\}, B = \{b\}))

- Standardized Platform with Independent Application Developer: A owns application a, B owns application b, joint ownership of platform, both firms have veto rights (S := (A = \{a, p\}, 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 := (A = \{a\}, B = \{b\})))
There are further possibilities of ownership structure which will not be considered here.⁵

Between stages 1 and 2 firms invest in the development of the applications and the platform. An application can benefit from the improvement of the platform by accessing its new API. At stage 2 firms renegotiate contracts. If they reach an agreement, changes to the platform are implemented and both firms have access to the new API. If negotiations break down, the owner of the platform can deny the other firm access to the new API, in case of a standardized platform the API is not allowed to be changed.

Revenues generated by the applications are determined by firms’ investments in the development of the platform \( p \) and applications \( a \) and \( b \), denoted as \( \pi, \alpha, \beta \in \mathbb{R}_0^+ \), respectively. We will assume that firms make non-verifiable investments in physical capital which can thus be fully expropriated by the asset owner.⁶ Firm A has the specific knowledge to make investments \( \alpha \) and \( \pi \), firm B has the specific knowledge for investment \( \beta \).

The only purpose of the platform is to increase applications’ revenues, it does not create revenues itself.⁷ 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 \((R_{aa} > 0, R_{a\pi} > 0, R_{b\beta} > 0, R_{b\pi} > 0)\)

---

⁵E.g. B may acquire A or one might give ownership of the platform to an independent firm.

⁶Physical capital implies hence that the investor does not gain bargaining power by 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.

⁷This 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.
and concave ($R_{\alpha\alpha} < 0, R_{\alpha\pi} < 0, R_{b\beta} < 0, R_{b\pi} < 0$) in investments. Further, we assume that an investment does not affect an other investment’s marginal revenue ($R_{\alpha\alpha} = R_{b\beta} = 0$). Because cross-derivatives are zero we will drop the unnecessary variable when writing the first derivative, e.g. $R_a(\pi) := R_a(\alpha, \pi)$.

In case of a break down of negotiations generated revenues are $r_a(A, B)(\alpha, \pi)$ and $r_b(A, B)(\beta, \pi)$, where $(A, B)$ 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 $r_a, r_{\alpha\alpha}, r_{\alpha\pi}, r_{b\beta}, r_{b\pi} \geq 0$, $r_{\alpha\alpha}, r_{\alpha\pi}, r_{b\beta}, r_{b\pi} \leq 0$, and $r_{\alpha\alpha} = r_{b\beta} = 0$ for all ownership structures. If negotiations break down joint revenues are always less or equal to the case of successful negotiations:

$$R_a + R_b \geq r_a(A, B) + r_b(A, B), \quad \forall \alpha, \beta, \pi, A, B.$$ 

## 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 firms get half of the surplus. We will consider ownership structures as if 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...

---

8We also need the technical assumptions $R_{kl}(0) \geq 1$ and $R_{kl}(\infty) < 1$ for all $k, l$.

9This terminology should be unambiguous as the question whether the platform is proprietary, standardized, or open source only matters when there is an independent application developer for this platform.
application development with a predetermined licensing of the platform.

3.1 Acquisition of B by A

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

\[ r^Q_a = R_a + R_b, \]
\[ r^Q_b = 0. \]

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

\[ V^Q_A = R_a + R_b, \]
\[ V^Q_B = 0. \]

3.2 Proprietary Platform with Independent Application Developer

In case negotiations break down, the same profit can be achieved with application \( a \). B, however, does not have access to the new API and his application has therefore lower revenues.

\[ r^P_a = R_a, \]
\[ r^P_b = (1 - \kappa)R_b, \quad \kappa \in (0, 1). \]

The surplus created by an agreement, \( \kappa R_b \), is split according to the Nash bargaining solution, thus ex post profits are

\[ V^P_A = R_a + \frac{\kappa}{2} R_b, \]
\[ V^P_B = \left(1 - \frac{\kappa}{2}\right) R_b. \]
3.3 Standardized Platform with Independent Application Developer

If negotiations break down, none of the firms can use the new API of the platform. Revenues are

\[
\begin{align*}
    r^S_a &= (1 - \lambda)R_a, \\
    r^S_b &= (1 - \mu)R_b,
\end{align*}
\]

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

The surplus is again split evenly and ex post profits are

\[
\begin{align*}
    V^S_A &= \frac{1}{2} (R_a - r_a + R_b - r_b) + r_a = \left(1 - \frac{\lambda}{2}\right) R_a + \frac{\mu}{2} R_b, \\
    V^S_B &= \frac{1}{2} (R_a - r_a + R_b - r_b) + r_b = \frac{\lambda}{2} R_a + \left(1 - \frac{\mu}{2}\right) R_b.
\end{align*}
\]

3.4 Open Source Platform with Independent Application Developer

Here no firm has a veto right, therefore, it is irrelevant whether an agreement is reached and

\[
\begin{align*}
    r^O_a &= R_a, \\
    r^O_b &= R_b.
\end{align*}
\]

Ex post profits are accordingly

\[
\begin{align*}
    V^O_A &= R_a, \\
    V^O_B &= R_b.
\end{align*}
\]

4 Investment Choice

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

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

\[
\max_{\pi, \alpha, \beta} R_a + R_b - \pi - \alpha - \beta.
\]
The first-order conditions

\[ R_{a\pi}(\pi^*) + R_{b\pi}(\pi^*) = 1, \quad (4.1) \]
\[ R_{a\alpha}(\alpha^*) = 1, \]
\[ R_{b\beta}(\beta^*) = 1, \]

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

In a second-best world, firms invest such that their investments maximize \( \text{ex ante profits} \)

\[ \max_{\pi,\alpha} \quad V_A - \alpha - \pi, \]
\[ \max_{\beta} \quad V_B - \beta. \]

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

**Acquisition**

\[ R_{a\pi}(\pi^Q) + R_{b\pi}(\pi^Q) = 1, \quad (4.2) \]
\[ R_{a\alpha}(\alpha^Q) = 1, \]
\[ \beta^Q = 0. \]

**Proprietary Platform**

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

**Standardized Platform**

\[ \left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi^S) + \frac{\mu}{2} R_{b\pi}(\pi^S) = 1, \quad (4.4) \]
\[ \left(1 - \frac{\lambda}{2}\right) R_{a\alpha}(\alpha^S) = 1, \]
\[ \left(1 - \frac{\mu}{2}\right) R_{b\beta}(\beta^S) = 1. \]
Open Source Platform

\[ R_{a\pi}(\pi^O) = 1, \quad R_{a\alpha}(\alpha^O) = 1, \quad R_{b\beta}(\beta^O) = 1, \quad (4.5) \]

4.1 Simple Setup

To be able to make statements about investment choice, we will have to make further assumptions. First, we will assume that the investments have the same influence on both application revenues, \( R_{a\pi}(\alpha) = R_{b\pi}(\beta) \). We further assume that having no access to the changed API has the same effect in all cases (\( \kappa = \lambda = \mu \)).

Investment levels are ranked as shown in Table 1. The proof is analogous to Proposition 1, which will come in the next subsection.

<table>
<thead>
<tr>
<th></th>
<th>( \pi )</th>
<th>( \alpha )</th>
<th>( \beta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-best (*)</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Acquisition (Q)</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Proprietary Platform (P)</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Standardized Platform (S)</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Open Source Platform (O)</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Ranking of investment levels for the simple case.

Here a standardized platform is dominated by both an open source and a proprietary platform. As we do observe standardization in reality, this result is unsatisfactory. Here we will provide two possible explanations why standardization can be optimal nevertheless. Either parameter values are different than assumed above (see next subsection). Or stage 0 makes standardization attractive. We will come back to this when discussing stage 0 (see particularly Subsection 5.2 and Section 6). Other explanations are given in Appendix A.

\[ ^{10} \text{Examples are the database query language SQL, Java with the Java Community Process, Microsoft with the standardization of parts of the .NET system, etc.} \]
4.2 Different Parameter Values

One possible explanation for standardization is that parameter values are different than initially assumed, e.g. $\lambda < \mu < \kappa$. We assume that in case of no agreement firms still benefit from improvements of the platform, that the effect of no agreement is less severe than in the case of an independent application developer, and that the developer of the platform loses less in case of a breakdown of negotiations.\footnote{The last assumption can be justified by the idea that knowledge is created by developing the platform has higher benefits for firm A in case that there is no agreement. This could maybe be modeled alternatively as investment in human capital.}

Proposition 1 states that investments are less or equal to first-best investments and gives a ranking of investment levels for different ownership structures. Table 2 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 $\pi$ we have for the left-hand-sides of Eqs. (4.1), (4.2), (4.3), (4.4), and (4.5)

\[
R_{a\pi}(\pi) + R_{b\pi}(\pi) > R_{a\pi}(\pi) + \frac{\kappa}{2} R_{b\pi}(\pi) > \left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi) + \frac{\mu}{2} R_{b\pi}(\pi) > R_{a\pi}(\pi). \tag{4.6}
\]

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,

\[
R_{a\pi}(\pi^*) + R_{b\pi}(\pi^*) = R_{a\pi}(\pi^Q) + R_{b\pi}(\pi^Q) = R_{a\pi}(\pi^P) + \frac{\kappa}{2} R_{b\pi}(\pi^P) = \left(1 - \frac{\lambda}{2}\right) R_{a\pi}(\pi^S) + \frac{\mu}{2} R_{b\pi}(\pi^S) = R_{a\pi}(\pi^O). \tag{4.7}
\]
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 $\pi (R_{app} < 0, R_{bpp} < 0)$.

ii) Analogously.

iii) Analogously. Additionally, $\beta^Q \leq \beta^P$ follows from $\beta^Q = 0$ and $\beta \in R^+_0$.

\[
\begin{array}{ccc}
\pi & \alpha & \beta \\
\hline
\text{First-best (*)} & 1 & 1 & 1 \\
\text{Acquisition (Q)} & 1 & 1 & 4 \\
\text{Proprietary Platform (P)} & 2 & 1 & 3 \\
\text{Standardized Platform (S)} & 3 & 2 & 2 \\
\text{Open Source Platform (O)} & 4 & 1 & 1 \\
\end{array}
\]

Table 2: Ranking of investment levels for the platform and applications a and b depending on ownership structure. A lower number means higher investment.

5 Asset Creation at Stage 0

We have so far 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 $\gamma$ 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: $R_b^0 > 0$, $R_b^\gamma < 0$, $R_b^{\gamma \gamma} = R_b^{\gamma \gamma} = 0$. $R_a$ is not affected by $\gamma$.

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 a proprietary
platform (P), 2. A standardized the platform at stage 0, B develops an application for a 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

\[ U^k_A := V^k_A - \alpha^k - \pi^k, \]
\[ U^k_B := V^k_B - \beta^k, \]

with \( k \in \{Q, N\} \). Assuming that negotiations are efficient, acquisition will happen if \( U^Q_A + U^Q_B > U^N_A + U^N_B \). In case of an acquisition the generated surplus is divided according to the Nash bargaining solution, both firms get \((U^Q_A + U^Q_B - U^N_A - U^N_B)/2\) additionally to the profit they would get without an acquisition. Therefore, A has the profit

\[ U^N_A + \frac{U^Q_A + U^Q_B - U^N_A - U^N_B}{2} = \frac{U^Q_A + U^N_A + U^Q_B - U^N_B}{2} \]

and B is bought for the price

\[ U^N_B + \frac{U^Q_A + U^Q_B - U^N_A - U^N_B}{2} = \frac{U^Q_B + U^N_B + U^Q_A - U^N_A}{2}. \]

### 5.2 Stage 0

At stage 0 firm A first commits to either nothing \((N = P)\), to a standardization of its platform \((N = S)\), or to open sourcing its platform \((N = O)\). Next B has to decide whether it wants to develop a first version of its application and if it does want to, whether as an independent developer or an employee of A. If B does not enter the market it earns its 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 its application, it makes a non-verifiable investment in physical capital \(\gamma\). 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 $U_B^{N\gamma} = 1$. 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 $\gamma$ is hence given again implicitly by $U_B^{N\gamma} = 1$. If $\gamma$ and $\beta$ were complements (i.e. $R_{b\beta\gamma} > 0$), $\gamma$ 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 a different firm 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. This 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 asset creation, the same applies as for $\beta$: $\gamma^* = \gamma^O \geq \gamma^S \geq \gamma^P \geq \gamma^Q$, 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

\[ E.g. \text{ for } N = P \text{ the derivative of } (U_B^Q + U_B^N + U_A^Q - U_A^N)/2 \text{ with respect to } \gamma \text{ is } \partial[(0 + (1 - \kappa/2)R_b + R_b - (\kappa/2)R_0)/2]/\partial \gamma \text{ which is equal to } U_B^{N\gamma}. \]

\[ \text{And if we modify the model such that } U_A \text{ increases in } \gamma, \text{ this can be beneficial for A even if there is no acquisition at stage 1.} \]
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 ($\gamma^{NQ}, N \in \{P, S, O\}$). 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 $(U_B^Q + U_B^N + U_A^Q - U_A^N)/2$ 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 ($\gamma^{PQ}$) in asset creation.

- **Standardized.** If higher investment in asset creation is needed, A commits to standardize his 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 $\gamma$ 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 devel-
oper 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 $\gamma^N$ and the newer $\beta^N$ version of his application according to the licensing of the platform ($N \in \{P, S, O\}$). Investment $\beta$ 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 $\gamma$ 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 he 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 non-entry can 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 firm $A$ owning both a platform and an application running on the platform. It would be important for the success of the platform if firm $B$ developed 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 his 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 $A$ 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 his platform. This can happen even if entry were optimal in first-best.

Facing the abovementioned alternatives, firms choose the least inefficient one.
Appendix

A On Standardization

These are alternative explanations for the existence of standards besides the explanations provided in the main text.

- open source software is often also free\(^\text{14}\) (for various reasons), this reduces open source revenues and can make open source less attractive than standardization. However, having an independent application developer for a proprietary platform is still more attractive than standardization.

- B’s marginal benefit from platform improvements is larger than A’s \(R_{b\pi} \geq R_{a\pi}\). Here standardization is better than open source, but a proprietary platform is still better.

- stochastic \(R_{b\pi}\) (realization in period 1). \(R_{b\pi}\) might have a negative realization, i.e. changes in the platform have a negative impact on the alternative application, i.e. compatibility of the platform and the alternative application are reduced (e.g. Windows and DR-DOS). Here it is a question what not revealing the API would mean for an independent developer (\(\kappa > 0, \kappa = 0?\)) and whether the independent developer could pay to block changes if they are not socially optimal. Standardization would be better, because blocking power for B would let A internalize some of the costs of incompatibility. Open source would let B use a previous version of the platform, hence no loss from incompatibility.

\(^{14}\)One often does not make the distinction between open source software and software which is given away for free. However, to be precise, open source means that the source code is accessible to the user, this does not exclude the possibility that he had to pay for it. On the other hand, software that comes free of charge may have source code not accessible to the user, and is hence not open-source.
Standardization could be explained by Rosenkranz and Schmitz (1999) (know-how disclosure and incomplete contracts).

References


