

# The Penguin Has Entered the Building: The Commercialization of Open Source Software Products

Andrea Fosfuri, Marco S. Giarratana, Alessandra Luzzi

Department of Business Administration, Universidad Carlos III de Madrid, 28903 Getafe, Madrid, Spain  
{fosfuri@emp.uc3m.es, marco.giarratana@uc3m.es, alessandra.luzzi@uc3m.es}

Previous literature on open source software (OSS) mostly analyzes organizational issues within communities of developers and users. This paper focuses on for-profit organizations that release software products under OSS licenses, and argues that variations in their endowments of intellectual property rights, namely patents and trademarks, help to determine which firms will tend to incorporate OSS into commercial products. We explain whether and under what conditions preexisting stocks of intellectual property rights can be useful complementary assets that allow firms to benefit directly or indirectly from commercializing OSS products, and test our hypotheses on a novel data set built on firms' announcements of OSS product releases in the specialized press between 1995 and 2003. We find three robust results: (a) firms with large stocks of software patents are more likely to release OSS products; (b) firms with large stocks of software trademarks are less likely to release OSS products; (c) firms with large stocks of hardware trademarks are more likely to release OSS products.

*Key words:* open source software; product introductions; patents and trademarks; complementary assets

*History:* Published online in *Articles in Advance* March 11, 2008.

## Introduction

Not only is open source software (OSS) currently gaining strength as a development process (Lerner and Tirole 2002), but it is also attracting increasing commercial interest among firms. The best known examples are IBM, Sun Microsystems, and Oracle, all of which have started to invest in and legitimize the use of Linux for enterprise applications (Koenig 2004). In the first quarter of 2005, Novell realized Linux-related product revenues of \$44 million USD, which accounted for 15% of its total sales in that period (*eWeek* 2005). Other firms, such as Red Hat and SCO Group, have specialized as OSS distributors, aggregating, integrating, and optimizing the newest software codes available from the OSS community.

Firms' increasing commercial interest in OSS constitutes an empirical puzzle insofar as it defies the traditional wisdom holding that a weak regime of appropriability seriously undermines firm incentives to release new products (Teece 1986). Thus, why do for-profit firms engage in the commercialization of OSS products? This article argues that firms can profit from their investment in OSS by relying on the control of complementary resources, and that the heterogeneity in the distribution of such resources explains why some firms take more OSS commercial actions than others.

OSS characteristics contrast sharply with the standard model for proprietary software, whereby innovative firms conceal their source code and use licenses to deprive users of the ability to share and modify the original software (Dam 1995). Instead, at the heart of any

current OSS product there is a license that both waives the principal rights assigned to the software creator by copyright law and grants users the right to access, modify, and redistribute the source code.<sup>1</sup> Software code that is revealed as OSS then becomes freely available, increasing the likelihood of imitation and substantially reducing the releasing firm's lead time and competitive advantage.

To be sure, firms can use different mechanisms to boost the appropriability of OSS products. For example, they can combine an OSS with a proprietary license (Hecker 1999, Henkel 2006). Alternatively, they can use traditional protection mechanisms such as legal rights and secrecy on at least part of the software, or even build an exclusive relationship with a committed developer community (Dahlander and Magnusson 2005). These alternative protection mechanisms notwithstanding, it is indisputable that OSS significantly reduces the level of code secrecy compared with proprietary software.

Our theoretical analysis starts with the observation that, over the last decade, OSS has reemerged as a mode for developing and organizing software innovation. Firms can respond to such a change in the software development process in different ways: by adjusting to, resisting, or supporting it. The other essential element of our analysis is the sensible reduction of the level of protection of the core product, which forces firms to secure the control of other, possibly complementary, resources to benefit from their commercialization efforts (Teece 1986). Several scholars point out how preexisting

assets, when protected and difficult to imitate, drive a firm's initial competitive position and affect its ability to adapt and exploit new opportunities (Tripsas 1997, King and Tucci 2002). Protected assets can be considered nonsubstitutable and costly to imitate resources (Barney 1991), meaning that variations in such protected assets could explain a large part of the differences in OSS commercialization strategies across firms. Specifically, we focus on the stocks of patents and trademarks that firms have accumulated before commercializing their first OSS product. These represent protected assets that are almost exogenous to the new conditions because they have been gathered under the proprietary software paradigm. We investigate whether and why the endowments of these property rights are (or are not) complementary to the OSS, which is the intellectual property in need of protection.

In this context, we formulate the following four research hypotheses: (a) firms with large stocks of software patents are more likely to release OSS products; (b) firms with large stocks of software trademarks are less likely to release OSS products; (c) firms with large stocks of hardware trademarks are more likely to release OSS products; (d) firms with large stocks of hardware patents are more likely to release OSS products.

We test these arguments by providing an econometric analysis on a unique data set of the commercialization of OSS products by for-profit firms. Our sample, built on the announcements of product introductions in the specialized press, tracks down all OSS product releases by firms from 1995 to 2003 in the two major software niches: operating systems and applications. This represents a novel data set that is both systematic and comprehensive. We use a count data model on the number of OSS product releases to test our hypotheses.

There is a fairly recent but growing body of literature on OSS. Previous academic research in organization and economics focuses mainly on understanding the motivations behind the participation of individual agents within the collective development process, their overall incentive structure, the allocation of rewards within OSS projects, and the sustainability of such a model for innovation (von Hippel and von Krogh 2003, Shah 2006, Roberts et al. 2006, O'Mahony 2003, O'Mahony and Ferraro 2007, Gambardella and Hall 2006). Traditionally, OSS has been considered a bottom-up movement, where most of the significant activity takes place in Internet-based communities of independent developers who collaborate with each other in creating software that they, or their organizations, need. As a consequence, both the theory and the empirical evidence on firms' commercialization of OSS products are scant. Although von Hippel and von Krogh (2003) argue that the free disclosure of source code limits direct paths to profiting from OSS products for commercial firms, they recognize that firms may still find indirect ways to appropriate the returns from investments in OSS. For example,

firms may benefit from patronizing OSS programs if these complement proprietary software or hardware that they already sell. Other authors emphasize that appropriability might not be firms' main concern if network externalities are important. OSS could increase the speed of diffusion of a given standard, and firms could then benefit by exploiting products that are complementary to that standard. Recent research based on survey data provides empirical evidence on the latter point (Henkel 2006, Gruber and Henkel 2006, Bonaccorsi et al. 2006). We contribute to this literature first by offering, to the best of our knowledge, the most comprehensive data on commercialization patterns in OSS, and second by providing a framework for understanding whether preexisting stocks of intellectual property rights can (or cannot) be useful complementary assets that allow firms to benefit directly or indirectly from using OSS code in commercial products.

The rest of the paper is organized as follows. Section 2 explains the theoretical framework on which our main hypotheses are based. Section 3 describes the data and the construction of the sample, the empirical methodology, and the variables used in the estimations. Section 4 shows the results, and §5 concludes with a discussion of the implications of our findings, some possible avenues for future research, and some of the limitations of this work.

## Theoretical Framework and Hypotheses

Our premise is that preexisting assets affect the direction as well as the pace of change and adaptation of a firm (Tripsas 1997, King and Tucci 2002). Compared with proprietary software, OSS entails important differences in the way products are developed. Firms might respond to such a change in the software development process in different ways: by adjusting, resisting, or supporting it. Thus, following the resource-based theory tradition, we investigate how heterogeneity in preexisting resource endowments among firms leads to variations in their response to a changing environment (Barney 1991)—in this case, differences in their efforts to commercialize OSS products.

We argued in the introduction that a salient difference between the OSS paradigm and the proprietary software paradigm is the level of protection of intellectual property, which is relatively lower in the former. Under the OSS paradigm, protection of software products through source-code secrecy is seriously undermined.

As the seminal work of Teece (1986) illustrated, when legal protection of intellectual property is weak, firms must rely on the control of complementary assets to profit from their product innovation efforts. To confer an advantage, such assets must be difficult for potential competitors to replicate. That is, even if the new product is easy to imitate, the bundle made up of the new product and the complementary assets must not be. We focus

our analysis on the protected intellectual property that a firm has accumulated before the advent of the OSS paradigm, which, by definition, is not easily imitable.

Because such protected intellectual property has been accumulated under the proprietary software paradigm, it is not clear a priori that it can be leveraged into the OSS paradigm. These assets are the result of several factors, including initial endowments, organizational learning, and path dependence, which are mostly exogenous to the new conditions (Dosi 1988). Hence, pre-existing assets could be complementary and thus drive a favorable initial competitive position for the firm, or they could be antonymous and seriously undermine the efficacy of adaptation (i.e., by generating organizational rigidities, as in Gatignon et al. 2002). Specifically, we consider two types of intellectual property protection, namely, patents and trademarks.

Patents are legal titles granting owners the exclusive right to make commercial use of their innovation and the enforcement power to prevent others from using it or to set the terms on which it can be exploited. Firms file patents to secure legal protection of their inventions—for instance, a new product—although patents also serve other purposes such as blocking rivals' research, preventing suits, reinforcing bargaining power in cross-licensing negotiations, sustaining technology transactions, etc. (Cohen et al. 2000, Ziedonis 2004).

In the software industry, a patent usually protects an algorithm, that is, a series of step-by-step procedures that are necessary to perform a task. If there are mathematical procedures beyond the algorithm, they are protected, too. Although software patents were granted in the United States before 1994, only after that date, thanks to a court decision, did software inventions become patentable *per se* and not only in conjunction with hardware inventions, so that any real difference between the treatment of software and that of other inventions was essentially eliminated by the U.S. Patent and Trademark Office (USPTO). The number of software patent applications increased dramatically after 1994. This has generated an intense debate between detractors and advocates of software patentability (see, for instance, Hall and MacGarvie 2006), which falls outside the scope of this paper. What matters for our argument is that firms differ in their stocks of software patents before they make inroads into commercializing OSS products.

Trademarks are combinations of “words, phrases, symbols, or designs that identify and distinguish the source of the goods or services” (USPTO Documentation, <http://tess.uspto.gov>). Firms can register as a trademark a new name, a jingle or a slogan, a new image, or a logo (e.g., “don't leave home without it”). In this way, they secure legal protection of their investment in marketing, reputation for quality, brand names, and distribution channels. Even if trademarks do not protect against the imitation of the product *per se*, they do help to boost

appropriability by securing control of these complementary assets.

Trademarks can protect not only the name and the logo of a product, but also promotions. “Pleasure for Windows,” “Smarter Retailing,” and “Global Access to Local Knowledge” are all trademarks registered by Microsoft. It is also common to protect the product name with several logos that differ in shape, design, and color. For example, by the year 2003, seven live trademarks protected the name of Nintendo's classic video game “Super Mario Bros.” The great importance of trademarks in the software industry is exemplified by the court battle that Microsoft waged against Lindows.com for the use of the trademark “Lindows,” which—according to Microsoft— infringed on several Microsoft trademarks. The court not only agreed to prohibit the use of that trademark, but also assigned a penalty of about \$400,000 USD to Lindows.com (*PR Newswire* 2003). Trademark owners pay different types of fees for each class of goods or services for which a trademark is registered, and they have to prove periodically that they are using the trademark in the relevant market; even if the owner is willing to pay the fees, a trademark is cancelled if it is not commercially used for five consecutive years after registration.

Whereas patents have received great attention in both management and economics literature, academic interest in trademarks has only recently emerged. Previous studies show that trademarks represent a good proxy for the products and markets in which a firm operates, and that they are correlated with sales and stock market value (Seethamraju 2003, Smith and Parr 2000).

Below, we explore how preexisting stocks of patents and trademarks affect efforts at commercializing OSS products.

### Software Patents

We highlight three reasons that help explain why a large portfolio of software patents can be a complementary asset that favors the commercialization of OSS products.

The first reason is the potential complementarity between patented software and OSS products. Arora (1995) demonstrates that firms can still extract rents from the transfer of know-how that is easily imitable—avoiding moral hazard and transaction cost problems—provided that such know-how is bundled with complementary knowledge that is well protected. Therefore, firms with large repositories of patented software algorithms have more chances to extract rents from OSS products by controlling complementary assets (Teece 1986). Some firms enclose in the transaction the payment of a royalty over the patented technology together with the services sold on top of the OSS product. A good example of this strategy is Squeezebox, an MP3 player. The server software that controls the device, SlimServer, is released under the common OSS

general public license (GPL) and has been developed with the contribution of a worldwide open source community. However, it is necessary to have a series of extra codecs installed to play back most audio formats, which are patent protected and not included in the standard installation (*Personal Computer World* 2006).

The second reason is strategic and relates to the possibility of controlling the development of an OSS project. Mann (2006) highlights how firms with software patents can direct the evolution of an OSS project in two ways. First, by threatening or exercising enforcement rights, they can make it prohibitively difficult for new contributors to obtain patents that write onto related knowledge. Second, they can discourage those contributions that pursue a deployment of the project that is too distant from the aims of the firm. For example, in workstations, Sun Microsystems controls the evolution of the Solaris open project, building a dense fence of patented technologies around it. Greater control over an OSS project enhances appropriability and thus reduces the costs of making the source code open. As a side effect, however, tight control of the OSS project might discourage participation of independent developers (Shah 2006) and could be detrimental, especially when projects are in their initial stages and require substantial contributions from the community.

The third reason is defensive. When a firm releases an OSS product that bundles several, sometimes dispersed, contributions, it is likely at risk of infringing on some patents held by other entities. However, the wider a firm's patent portfolio, the stronger its bargaining power, and the higher its chance to avoid litigation or to establish friendly agreements with potential litigators. For example, in semiconductors, Hall and Ziedonis (2001) and Ziedonis (2004) show that when the markets for technologies are fragmented, large patent portfolios help firms lower transaction costs and resolve hold-up problems through the use of cross-licensing agreements. Thus, our first hypothesis is

**HYPOTHESIS 1.** *The larger a firm's preexisting stock of software patents, the more likely it is to build OSS into its commercial software products.*

### **Software Trademarks**

Firms with large stocks of software trademarks have invested heavily to protect their marketing efforts, brands, reputation for quality, and distribution channels. This investment is tailored to the proprietary software paradigm, and we argue that it cannot be easily leveraged into OSS.

As Mitchell and Singh (1992) have argued, the fear of product cannibalization prevents firms from investing in emerging and new markets. Typically, profits from new products are uncertain, and a firm is relatively unwilling to jeopardize existing, stable income streams by making

risky investments (Conner 1998). The industrial organization tradition predicts that an incumbent has fewer incentives to introduce new products because they erode profits in existing lines of business (Ghemawat 1991). Thus, the cannibalization effect due to the commercialization of OSS products is likely to be greater for those firms with larger stakes in proprietary software.

Moreover, a firm that has established and protected a reputation in proprietary software has done so to increase the consumer's willingness to pay a premium for quality, reliability, and brand name. It therefore has an interest in emphasizing the costs of switching brands. OSS requires a quite different strategic positioning. First, many customers/consumers of OSS products are more cost conscious (Von Hippel and Von Krogh 2003). Second, OSS products tend to be priced much lower than equivalent proprietary products. Simultaneously offering products with a different positioning inside a unique brand strategy might lead to cannibalizing investment in proprietary software brand names and reputation (Chaudhuri and Holbrook 2001).

In principle, a strong brand name or reputation for quality can be an intangible asset that is not easily imitable and is exploitable in other domains. Software trademarks could be complementary to OSS products and could boost their appropriability, much as patents do. However, unlike patented technologies, the successful transfer of existing trademarks into a different context greatly depends on the customers' reaction. Firms rely heavily on brand investments to convince the consumer that their products are superior to those of competitors. Usually this occurs in market niches in which a variety of products with similar performances are available. Under these circumstances, entering OSS might be harmful because it might suggest to the consumer that competing products are nearly identical.

Thus, firms with large software trademark portfolios will be more likely to adopt wait-and-see strategies.

**HYPOTHESIS 2.** *The larger a firm's preexisting stock of software trademarks, the less likely it is to build OSS into its commercial software products.*

### **Hardware Trademarks**

Although proprietary software and OSS can, broadly speaking, be considered alternatives, either open source or proprietary software is complementary to the hardware, that is, the machine on which the software runs. If hardware and software are characterized collectively as a value chain, then commoditizing those portions of the chain in which the company does not have a core competence helps earn higher returns from those portions of the chain in which it can compete. Firms that have invested heavily in establishing a brand name and reputation for quality in hardware will benefit if OSS either reduces software production costs (in case the firm typically develops in-house the software for its machines)

or minimizes hold-up from software suppliers (in case the firm typically outsources software production).

The costs of assembling an OSS product are much lower than those of developing new software from scratch (Samuelson 2006). Firms can combine various available OSS modules at only a fraction of the cost of developing ex novo equivalent proprietary modules (*Economist* 2005). Von Krogh et al. (2005) find software reuse quite common among 15 OSS projects; Spaeth et al. (2006) show that, in the OSS Debian project, software reuse follows a power law distribution, with some software chunks intensively reused. OSS also opens up the opportunity for reducing the bargaining power of specialized suppliers of proprietary software by offering a cheaper and more customizable alternative to their products. Firms with large stocks of hardware trademarks therefore have an incentive for backing OSS initiatives from which they will eventually benefit as a result of complementarity. Industry jargon refers to this as patronizing strategy.

It is also worth noting that firms with a strong brand name and reputation in hardware are less concerned about protecting their software. First, if software is used simply to make hardware machines operative, such firms prefer OSS because by making the source code public they are relieved of the burden of servicing and updating the software. Second, when the OSS solution is customized to the machine, imitation of the software program becomes more difficult because the process of customization entails a great deal of tacit and nonobservable knowledge.

These arguments lead to our third hypothesis.

**HYPOTHESIS 3.** *The larger a firm's preexisting stock of hardware trademarks, the more likely it is to build OSS into its commercial software products.*

### Hardware Patents

The arguments supporting Hypothesis 3 apply almost directly to a firm's stock of hardware patents. Firms that have invested in protecting their hardware technology benefit from the commoditization of software due to the presence of potential complementarities between hardware and software knowledge bases. In addition, as discussed in Hypothesis 1, firms can better profit from their OSS products by exploiting the complementarity with the hardware on which they exert greater legal control through patents (Arora 1995).

Thus, our last hypothesis is

**HYPOTHESIS 4.** *The larger a firm's preexisting stock of hardware patents, the more likely it is to build OSS into its commercial software products.*

## Data and Methodology

### Sample Construction

Our sample is composed of all the firms that announced through the specialized press the introduction of a software package based on an OSS license between 1980 and 2003. We select our data from the databases Infotrac's General Business File *ASAP* and *PROMT*, by searching for press articles that report a "product announcement," a "new software release," or a "software evaluation" in the software sector (Standard Industry Classification (SIC) code 7372) and that contain in their text the words "open source" or "Linux."<sup>2</sup> After reading the text of each article, we select only those events that clearly refer to a product that is described in the article as being "open source," theoretically excluding all proprietary software products running on OSS platforms. We then extract from each article the name of the company and the date of product introduction (month and year). We find that the first commercial announcement of an OSS product was in May 1995. We cannot measure directly the level of vaporware in our product introduction notifications—a well known limitation in news event data—but we believe that the high integrity of the journals from which the announcements are collected limits the likelihood of its presence in our data (*eWeek*, *Computer Reseller News*, *Electronic Engineering Times*, and *PC Magazine* account for more than 70% of the announcements). In addition, 65% of the selected articles also contain a product evaluation, which implies that the journal's reviewer has at least seen the product prototype. Our robustness checks show that focusing only on these articles does not qualitatively change our findings. Although we do not have precise information about the type of license that is used for each product, we can identify all products described in the articles as open source and Linux-based. Such products are licensed under the GNU-GPL scheme, which imposes the burden of reciprocity, i.e., the source code of derived work must be made available to all receivers of the software. Note that these products represent more than 80% of our whole sample of commercially released OSS products. We take this feature into consideration when we perform some robustness checks.

Using the SIC codes reported in the articles, we divide the OSS products into two main niches: operating systems (OPSYS; SIC codes 737261 and 737250) and applications (APP; six digit SIC codes beginning with 7372, other than -61 and -50). When the SIC code is not available at six digits, we analyze the text of the article to distinguish between the two niches. The literature provides evidence of potential differences between these niches in the factors that drive product releases and determine their success. For operating systems, these factors are the strength of network externalities and their performance on hardware; for applications, they

**Table 1** Distribution of Firms and OSS Product Releases from 1995 to 2003 by Sector of Core Activity

	OPSYS			APP		
	Number of firms	Av. size (employees)	Number of products	Number of firms	Av. size (employees)	Number of products
Electronics	13	42,827	17	7	34,840	7
Hardware	77	11,390	286	30	24,985	83
Software	195	1,155	490	154	1,368	236
Semiconductors	18	9,451	61	2	39,434	5
Telecommunications	12	13,185	17	11	31,458	19
Other	5	283,190	6	9	6,586	10
Total	320			213		
Number of products			877			360

Source. Our elaborations are from Infotrac's General Business File ASAP and PROMT databases and from Bureau van Dijk's Icarus, Amadeus, and Jade databases.

are the features of the products, the marketing approach, and the speed of customer learning-by-using (Torrizi 1998). Moreover, earlier studies of the software industry (Gandal et al. 1999) demonstrate that the diffusion of an operating system standard is intrinsically associated with the availability of a large portfolio of applications that could be run on that software base.

We check for the group structure of our firms by using information from the Business and Company Resource Center database, Gale Group's Infotrac. We end up with 213 different entries in APP, which account for 360 different products, and 320 entries in OPSYS, which account for 877 products. Because 72 firms entered both niches, the total number of entrants in the OSS market is 461. All company data used in our analysis are obtained from Bureau van Dijk's databases Icarus, Amadeus, and Jade for American, European, and Asian firms, respectively. Table 1 shows the distribution and average size of the firms in our sample by core sector of activity. Note that 85% of the firms that introduced OSS products in OPSYS (and 86% of those that introduced OSS products in APP) are either software or hardware firms, and that these firms account for 88% of OSS product releases in OPSYS (and 89% in APP).

### Estimation Procedure and Dependent Variable

We estimate a count data model where the dependent variable is the number of new OSS products (or new versions of already launched OSS products) announced by a firm from the date of its first OSS product release until December 2003. We believe this variable captures well the efforts a firm makes to build OSS into its commercial software products. Moreover, the literature has suggested that in a system with free disclosure of technical knowledge, such as OSS, firms must continually introduce new products or new updated versions to survive and gain a competitive advantage (Garud and Kumaraswamy 1993).

Table 2 shows the distribution of the number of OSS product introductions along with the average presence of a firm in the market (in months) for groups of firms classified by the number of products they introduced.

Of the firms, 36% in OPSYS and 21% in APP released more than one product before December 2003. Most of the multirelease firms introduced, on average, one product per year. Among firms with more than one release, 17.3% of those in OPSYS released between 2 and 12, and 22.2% of those in APP released between 2 and 6. It is worth noting that the majority of those firms that had introduced only one product were late entrants in the market (about three years of presence on average in both niches). Accordingly, as we report in the control variables section, we introduce into our estimations a firm's duration of presence in the OSS market.

We hypothesize that the number of OSS product releases is generated by the function  $y = f(x, \beta)$ , where  $y$  is product count,  $x$  is the set of explanatory variables, and  $\beta$  is a vector of parameters to be estimated. One econometric challenge is in the fact that our sample is not randomly selected because we observe only those firms that have introduced at least one OSS product during the period under scrutiny. Thus, we cannot directly use the Poisson (or negative binomial) model for count data, and we need to run an estimation procedure that incorporates the correction for the no-release observations. We therefore use a procedure based on

**Table 2** Distribution of the Number of OSS Product Releases by Firms

Number of products	OPSYS			APP		
	Number of firms	%	Months on the market	Number of firms	%	Months on the market
1	205	0.641	35	168	0.789	39
2–5	90	0.281	45	38	0.178	47
6–10	14	0.044	51	3	0.014	53
11–25	6	0.019	62	3	0.014	56
>25	5	0.016	69	1	0.005	59

Source. Our elaborations are from Infotrac's General Business File ASAP and PROMT databases.

Notes. The mean of months on the market has been computed as the mean of the number of months between firm entry (month, year) and December 2003, where entry refers to a firm's first OSS product introduction.

Terza (1998), who shows how the number of product introductions conditional on a selection equation can be estimated by nonlinear least squares, even if the actual conditional distribution is unknown.

We build a sample of firms that did not release OSS products during the period under study. These firms are randomly selected using as matching criteria the precise country (Japan, Germany, United Kingdom, Canada, and so on) and sector (three digit SIC code) composition of the original sample, so that the total dimension of the original sample is replicated for the no-release sample in both APP and OPSYS. We again use the databases Icarus, Amadeus, and Jade to select these firms.

Considering both the firms in the original sample and the no-release firms, we assume that the observed number of product introductions  $y_i$  follows a Poisson distribution conditional on a normally distributed error term  $y_i | \varepsilon_i \sim \text{Poisson}(\lambda_i)$ , so that its log-conditional mean can be modeled as a linear function of our covariates and controls,  $x_i$ , plus the error term  $\varepsilon_i \sim N(0, \sigma^2)$ :

$$\ln E(y_i | x_i, \varepsilon_i) = \ln \lambda_i | \varepsilon_i = \beta' x_i + \varepsilon_i.$$

The normal error term in this specification allows us to take into account unobserved heterogeneity, which is not modeled in the standard Poisson specification where the mean of  $y_i$  is equal to the variance and all of the heterogeneity is accounted for in the vector of covariates  $x_i$ .<sup>3</sup>

The estimation procedure is then run in two steps, where the former models a sample selection equation and the latter models the mean of the number of product introductions conditional on at least one product introduction. We address the first-step estimation through a Probit specification, where the dependent variable is a dummy equal to 1 if a firm has introduced an OSS product and 0 otherwise ( $d_i = 1$  if  $z_i^* = \alpha' w_i + u_i \geq 0$ ;  $u_i \sim N(0, 1)$ ). Given the first-step estimation of the parameters  $\alpha$ , the mean of the number of product introductions conditional on a product release ( $d_i = 1$ ) can be specified as (Terza 1998)

$$E[y_i | x_i, d_i = 1] = \exp(\beta^0 + \beta_1 x_{1i} + \dots) \left( \frac{\Phi(\alpha' w_i + \theta)}{\Phi(\alpha' w_i)} \right),$$

where  $\Phi$  is the cumulative function of a standardized normal distribution, the constant term  $\beta^0$  is equal to  $\beta_0 + \sigma^2/2$ , and  $\theta$  is the covariance of the bivariate normal distribution of the two error terms  $(\varepsilon_i, u_i)$ ,  $\theta = \rho\sigma$ . Even if the actual distribution of  $y_i | x_i, d_i = 1$  is unknown, the parameters  $\beta$  and  $\theta$  in the conditional mean function of the second step can be estimated by nonlinear least squares.

Because  $\hat{\alpha}' w_i$  is used in place of the unobserved  $\alpha' w_i$  in the estimates of the second step, the asymptotic covariance matrix for the two-step estimator must be adjusted after estimation. We use a consistent estimator for the asymptotic covariance matrix based on Murphy and Topel (1985) (see Green 2000, chap. 20).

## Main Independent Variables: Patents and Trademarks

*Software and Hardware Patents.* We count the number of unexpired software and hardware patents granted to a given firm up to the month of the announcement of its first OSS product. We download patents data from the USPTO database available at <http://tess.uspto.gov>. To distinguish between software and hardware patents, we apply search algorithms to the front page of the patent (see the appendix for a detailed explanation of these search algorithms). We label these variables *patentsoftware* and *patenthardware*.

*Software and Hardware Trademarks.* Like patents records, trademarks data are downloaded from the USPTO database available at <http://tess.uspto.gov>. To distinguish between software and hardware trademarks, we apply to the front page of the trademark search algorithms analogous to those applied to patents (see the appendix). We then create two variables: *trademarksoftware* and *trademarkhardware*, the preentry number of live USPTO trademarks in software and in hardware, respectively, filed by a firm up to the month before its first OSS product announcement.

Note that firms display important heterogeneity in their stocks of patents and trademarks, which suggests that these measures capture two different aspects of a firm's preentry position. Thus, it is not uncommon that a firm with a large stock of software trademarks has only few software patents granted by the USPTO or vice versa. For example, Oracle, Cisco Systems, and Computer Associates have trademark–patent ratios of 2, 3, and 27, respectively. Conversely, IBM's patent–trademark ratio is approximately 6. At the sector level, hardware and semiconductor firms have a higher propensity to patent software than to file software trademarks. Software firms show a higher propensity to file software trademarks than to patent. Details are available from the authors upon request.

## Controls

We include a set of controls in our estimations. First, we control for firm size and age, where *size* is measured as the logarithm of the number of employees in the year of the announcement of the first OSS product and *age* is proxied by three period dummies that are equal to 1 if the founding year of a firm is between 1976 and 1985 (*age76–85*), between 1986 and 1994 (*age86–94*), and after 1995 (*age95*). The baseline is a firm founded before 1976. These time breaks broadly correspond to the main development eras in the computer industry (workstation, personal computer, open source software). *Size* and *age* are standard controls in the entry and survival literature, and can be interpreted as proxies for a firm's scale and experience. We also control for the degree of firm diversification in the year of the announcement of the first

OSS product. We measure this by a Herfindahl index calculated over the total stock of a firm's trademarks distributed across the 11 different classes of the USPTO International Classification of Goods and Services. Note that the higher the index, the less diversified the firm. Thus, we call this variable *specialization*.

Second, given that we run separate estimations for the two niches (APP and OPSYS), and that 72 firms have announced product introductions in both of them, we control for the fact that firms that have already entered one niche may have a significantly different number of OSS releases in the other. Therefore, when we run our estimations for OPSYS, we introduce a dummy variable that is equal to 1 if a firm has introduced a software application before (and not after) its first operating system introduction, and zero otherwise (*anteapplication*). Similarly, in the estimations for APP niche, we create the dummy variable *anteopsystem*, which is equal to 1 if a firm has introduced an operating system product before (and not after) its first introduction of an application. The relevance of these control dummies is suggested by previous studies that find positive and strong feedbacks between applications and operating systems. Comparing DOS and CP/M products, Gandal et al. (1999) highlight that applications make operating systems more valuable and vice versa. These findings suggest that a firm that has entered into both niches could benefit from some sort of advantage in releasing new software products.

Third, a potential concern about the use of stocks of patents and trademarks is that there is significant heterogeneity in patent and trademark propensity across different sectors, as we also pointed out above. Software firms, for instance, have traditionally patented less than other firms in part because software per se could not be patented before 1994. Thus, our core variables might capture a sector effect rather than a firm effect. We take this into account by introducing five sector dummies: *hardware* (SIC code 357), *software* (737), *electronics* (36 except for 367 and 366), *semiconductors* (367), and *telecommunications* (366). In addition, we introduce two geographical dummies for European and Asian firms (*Europe* and *Asia*), considering American firms as the baseline.

Fourth, because firms announced their first OSS product at different times and we observed their product introductions only until December 2003, it is important to control for a firm's duration of presence in the market (Hardin and Hilbe 2001). We build the variable *timeexposure*, defined as the number of months elapsed between a firm's first OSS product introduction and the end of the period (December 2003). We insert the log of this variable in the second-step estimation with the coefficient constrained to 1. This is equivalent to adding a multiplicative factor  $TE_i$  to the conditional mean

$$E[y_i | x_i, d_i = 1] = TE_i \cdot \exp(\beta^0 + \beta_1 x_{1i} + \dots) \begin{pmatrix} \dots \\ \dots \end{pmatrix},$$

**Table 3 Descriptive Statistics**

	OPSYS (n = 320)	APP (n = 213)
<i>Patentsoftware</i>	12.406 (80.428)	16.845 (97.999)
<i>Trademarksoftware</i>	7.031 (21.252)	9.587 (25.629)
<i>Trademarkhardware</i>	4.637 (14.610)	5.727 (17.662)
<i>Patenthardware</i>	16.347 (109.991)	21.760 (132.811)
<i>Size</i>	10,636 (82,741.864)	7,927.07 (31,158.476)
<i>Age76–85</i>	0.206 (0.405)	0.221 (0.416)
<i>Age86–94</i>	0.322 (0.468)	0.380 (0.487)
<i>Age95</i>	0.35 (0.478)	0.277 (0.449)
<i>Specialization</i>	0.821 (0.278)	0.791 (0.294)
<i>Anteopsystem</i> ♦		0.188 (0.391)
<i>Anteapplication</i> ♦	0.069 (0.253)	
<i>Dummy_ele</i>	0.041 (0.198)	0.033 (0.179)
<i>Dummy_soft</i>	0.609 (0.489)	0.723 (0.449)
<i>Dummy_hd</i>	0.241 (0.428)	0.141 (0.349)
<i>Dummy_tlc</i>	0.037 (0.190)	0.052 (0.222)
<i>Dummy_sem</i>	0.056 (0.231)	0.009 (0.097)
<i>Dummy_eu</i>	0.141 (0.348)	0.127 (0.333)
<i>Dummy_asia</i>	0.069 (0.253)	0.042 (0.202)
<i>Timeexposure</i>	41.656 (16.618)	41.958 (20.733)

Source. Our elaborations are from Infotrac's General Business File ASAP and PROMT databases and from Bureau van Dijk's Icarus, Amadeus, and Jade databases.

Notes. Means are shown with standard deviations in parentheses.

♦Dummy variables.

indicating the relative time of presence of each firm. For the no-release firms, all variables are measured at December 2003.

Table 3 provides the basic descriptive statistics for the independent variables and controls, whereas Table 4 reports pairwise correlations.

## Results

Table 5 shows the results of the second-step, nonlinear least squares estimation, that is, the effect of our variables on the number of OSS product introductions. We discuss, first, the effect of a firm's preexisting trademarks and patents on the number of product introductions in OPSYS. Next, we show that some differences are observed in APP, and we propose a tentative explanation for this finding.

A firm's larger stock of preexisting software patents enhances the number of OSS product releases. Thus, as our first hypothesis conjectured, software patents are complementary assets whose control increases the appropriability of OSS products (Arora 1995). This finding also suggests that software patents can be used as bargaining chips to reduce potential hold-up problems and/or to allow the firm to better control the evolution of an OSS project (Ziedonis 2004, Mann 2006). Thus, software patents, which may represent intellectual property protection on other products or protection of components of a product also containing some OSS code, are playing an important role even within the OSS paradigm. Holding all other variables at their mean value, our estimated model predicts that doubling the number of software patents with respect to the mean of *patentsoftware*

**Table 4 Correlation Matrix**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 <i>Patentsoftware</i>	1																	
2 <i>Trademarksoftware</i>	0.54	1																
3 <i>Trademarkhardware</i>	0.55	0.52	1															
4 <i>Patenthardware</i>	0.58	0.54	0.57	1														
5 <i>Size</i>	0.22	0.24	0.26	0.28	1													
6 <i>Age76–85</i>	0.09	0.12	0.00	−0.01	−0.03	1												
7 <i>Age86–94</i>	−0.10	−0.12	−0.07	−0.07	−0.06	−0.29	1											
8 <i>Age95</i>	−0.17	−0.17	−0.09	−0.09	−0.09	−0.40	−0.49	1										
9 <i>Specialization</i>	−0.15	−0.11	−0.05	−0.06	−0.06	−0.03	0.04	0.11	1									
10 <i>Anteopsystem</i>	0.26	0.22	0.26	0.25	0.06	0.06	−0.01	−0.05	0.08	1								
11 <i>Anteapplication</i>	0.23	0.17	0.23	0.23	0.10	0.05	−0.09	−0.05	0.05	0.25	1							
12 <i>Dummy_ele</i>	0.14	0.16	0.03	0.06	0.11	−0.08	−0.01	−0.08	−0.03	0.01	0.01	1						
13 <i>Dummy_soft</i>	−0.14	−0.25	−0.11	−0.13	−0.15	−0.04	0.11	0.12	0.06	0.06	0.04	−0.25	1					
14 <i>Dummy_hd</i>	0.07	0.17	0.10	0.09	0.00	0.07	−0.08	−0.04	−0.04	−0.05	−0.02	−0.11	−0.70	1				
15 <i>Dummy_tlc</i>	0.09	0.06	0.03	0.04	0.01	−0.10	0.01	0.00	−0.02	−0.04	−0.01	−0.04	−0.25	−0.11	1			
16 <i>Dummy_sem</i>	0.00	0.03	0.02	0.04	0.01	0.08	−0.07	−0.04	−0.01	−0.05	−0.09	−0.05	−0.31	−0.14	−0.05	1		
17 <i>Dummy_eu</i>	−0.08	−0.07	−0.04	−0.04	−0.03	0.02	0.07	−0.07	0.03	−0.08	−0.03	−0.04	0.09	−0.10	0.06	−0.02	1	
18 <i>Dummy_asia</i>	0.07	0.13	0.02	0.05	0.07	−0.04	−0.05	−0.06	−0.05	−0.02	−0.01	0.23	−0.12	0.08	−0.02	−0.04	−0.11	1

would increase the number of OSS product introductions by 47%.<sup>4</sup>

In line with our second hypothesis, the larger the number of software trademarks, the lower the expected number of OSS product introductions in OPSYS. Thus, the threat of cannibalization plays an important role for those firms that have invested heavily in proprietary software brand name and reputation for quality. Holding all

other variables at their mean value, doubling the number of software trademarks with respect to the mean of *trademarksoftware* would reduce the number of OSS product introductions by 11%.

The number of hardware trademarks shows a positive and significant effect on the number of product introductions in OPSYS. This finding indicates that there are complementarities between hardware and software,

**Table 5 Nonlinear Least Squares Estimation (Second-Step Estimation; Corrected Standard Errors in Parentheses)**

	OPSYS		APP	
	(1)	(2)	(1)	(2)
<i>Patentsoftware</i> (Hypothesis 1)	0.012** (0.002)	0.012** (0.003)	0.003** (0.001)	0.002** (0.0008)
<i>Trademarksoftware</i> (Hypothesis 2)	−0.021** (0.005)	−0.012** (0.005)	−0.029** (0.009)	−0.017** (0.005)
<i>Trademarkhardware</i> (Hypothesis 3)	0.041** (0.006)	0.029** (0.009)	0.024** (0.009)	0.014* (0.007)
<i>Patenthardware</i> (Hypothesis 4)	−0.008** (0.002)	−0.007** (0.002)	−0.002 (0.002)	−0.001 (0.001)
<i>Size</i>	0.124* (0.070)	0.107 (0.708)	0.232** (0.062)	0.215** (0.040)
<i>Age76–85</i>	−0.220 (0.336)	0.193 (0.427)	−0.245 (0.423)	0.011 (0.261)
<i>Age86–94</i>	0.655* (0.360)	0.732* (0.396)	−0.428 (0.384)	−0.238 (0.297)
<i>Age95</i>	−0.028 (0.361)	−0.045 (0.437)	−0.164 (0.379)	0.065 (0.320)
<i>Specialization</i>	−0.229 (0.685)	−0.611 (0.809)	0.278 (0.446)	−0.113 (0.313)
<i>Anteopsystem</i> <sup>♦</sup>				0.517** (0.112)
<i>Anteapplication</i> <sup>♦</sup>		−1.121** (0.499)		
<i>Sector</i> <sup>♦</sup>		Yes		Yes
<i>Country</i> <sup>♦</sup>		Yes		Yes
<i>Timeexposure</i> <sup>a</sup>		Yes		Yes
Beta0	−4.533** (1.232)	−3.722** (1.600)	−4.570** (0.909)	−5.084** (0.819)
Theta <sup>b</sup>	0.304 (0.622)	0.079 (0.962)	−1.136** (0.333)	−0.683** (0.255)
Log likelihood	−845.864	−839.474	−333.586	−324.190
No. of observations	320	320	213	213

\*,\*\*Significant at \*10%, \*\*5%.

<sup>♦</sup>Dummy variables.

<sup>a</sup>Time of exposure is defined as the number of months that elapsed between firm entry (first OSS product release) and the end of the period (December 2003). It is used as a multiplicative factor to the conditional mean for the second-step estimations, i.e., it is used in log with the coefficient constrained to 1.

<sup>b</sup>The significant estimate for the covariance term  $\theta$  in the Applications niche reflects the presence of unobserved factors that are common to the two decisions of a firm (whether to enter the OSS market and how many products to launch). Its negative sign shows that the average number of product introductions would have been overestimated without a control for selectivity.

and that firms that have invested in establishing a brand name and reputation in hardware can benefit from commoditizing the software counterpart. Keeping all other variables at their mean value, doubling the number of hardware trademarks with respect to the mean of *trademarkhardware* would increase the number of OSS product introductions by 16%. This finding lends support to Hypothesis 3.<sup>5</sup>

Finally, the negative coefficient of *patenthardware*—significant only for the operating systems niche, and not significant in some of the robustness checks we describe—does not support our last hypothesis that legal protection of hardware components helps to boost the appropriability of OSS products. As a possible explanation for this finding, we conjecture that firms with strong technological specialization in hardware might have fewer incentives to invest in software products due, for instance, to fundamental differences in programming languages between hardware and software engineers.

The results are similar for APP. Of particular interest in this latter niche is the positive and highly significant coefficient of the control variable *anteosystem*, which suggests that a key driver for the number of applications released is whether a firm has previously launched an operating system. We interpret this finding as evidence of the patronizing strategy that some firms pursue: By backing OSS applications, firms increase the value of OSS operating systems. This picture is consistent with the initial stage of the OSS market, where—lacking an established and reliable set of application producers—firms that want to launch operating systems are in some way forced to release in-house applications to increase the value and legitimacy of their OSS operating systems.

As to the other control variables, *size* has a positive coefficient in APP, but is not significant in OPSYS, implying that the effects of scale are more important for the applications. Age dummies and *specialization* do not show significance, suggesting that experience and diversification competences play a limited role.

### Robustness Checks

We perform various robustness checks to validate our findings. The results of these alternative regressions are shown in Table 6. First, to ensure that our results are robust regardless of the type of license scheme, we run the regression only with product introductions that are both open source and Linux-based (Model 1 in Table 6). OSS products can be licensed under different schemes that allow for different levels of openness of the code. All Linux-based OSS products are licensed under the same GNU-GPL scheme, which imposes the burden of reciprocity. As we mentioned, these products represent 87% and 88% of our sample of OSS product introductions in OPSYS and APP, respectively. Second, we perform the regression with only those product announcements (about 65% of our whole sample) that

come from articles containing a product evaluation, which implies that the journal's reviewer has at least seen the product prototype. This is likely to reduce the problem of vaporware we mentioned (Model 2 in Table 6). Third, we run the regression only for those products that have both received an evaluation and are licensed as GNU-GPL (Model 3 in Table 6). Fourth, to check whether our findings are driven by outliers, we exclude the three largest firms in terms of protected intellectual property assets (IBM, Sun, and HP) from the estimation (Model 4 in Table 6). The estimated coefficients remain fairly stable in all these robustness checks. One result that is not robust is the significance of the coefficient of *trademarkhardware* for APP. We believe that this finding is consistent with the fact that applications need fewer linkages and synergies with hardware. Finally, Model 5 in Table 6 shows the results with our patent and trademark variables in log scale to check if estimates are biased because of forced increasing returns to scale. The results are qualitatively unchanged.

We also performed the following unreported robustness checks, which are available upon request: (1) we used *age* as a continuous variable; (2) we ran our regressions only for U.S. firms because trademark and patent variables are drawn from the U.S. Patent Office and, therefore, are measured more precisely for this subset of firms; (3) we pooled all product releases regardless of the niche to which they belonged; (4) we scaled patent and trademark counts directly by size; (5) we sequentially excluded our core independent variables from the estimation to test for multicollinearity biases. In all of these cases, the results remained broadly consistent.

### Discussion and Conclusions

Until recently, the OSS movement has been championed by individuals, mostly software engineers and developers, with little firm participation. However, anecdotal evidence suggests that firms have begun to take an increasing commercial interest in OSS opportunities. This paper accomplishes two goals. First, it provides novel and comprehensive data on the commercialization of OSS products. Second, it identifies some sources of heterogeneity in OSS product introductions across firms: Variations in preexisting stocks of intellectual property rights, namely patents and trademarks, help to explain why some firms are taking more commercial actions within the OSS paradigm than others. We have confirmed our hypotheses for a data set built on firms' announcements of OSS product releases in the specialized press between 1995 and 2003.

Three results are worth summarizing. First, we find that patents play an important role in appropriating the returns from the commercialization of OSS products. Firms with large stocks of software patents introduce more OSS products. Our second finding points to the

**Table 6 Robustness Checks**

	Model 1		Model 2		Model 3		Model 4		Model 5	
	OPSYS	APP								
<i>Patentsoftware</i> (Hypothesis 1)	0.011** (0.003)	0.003** (0.001)	0.015** (0.003)	0.003** (0.001)	0.014** (0.004)	0.004** (0.000)	0.011** (0.004)	0.003* (0.001)	0.499* (0.280)	0.159** (0.048)
<i>Trademarksoftware</i> (Hypothesis 2)	-0.018** (0.006)	-0.016* (0.009)	-0.018** (0.007)	-0.016** (0.004)	-0.021** (0.008)	-0.017* (0.009)	-0.009** (0.002)	-0.007** (0.003)	-0.756** (0.237)	-0.351** (0.072)
<i>Trademarkhardware</i> (Hypothesis 3)	0.031* (0.011)	0.018 (0.012)	0.034** (0.012)	-0.000 (0.016)	0.033** (0.012)	-0.019 (0.012)	0.031** (0.008)	0.004 (0.012)	0.451** (0.197)	0.149** (0.044)
<i>Patenthardware</i> (Hypothesis 4)	-0.006** (0.002)	-0.002 (0.002)	-0.009** (0.002)	-0.000 (0.001)	-0.008** (0.003)	-0.002 (0.002)	-0.008** (0.002)	-0.005 (0.005)	0.146 (0.260)	0.018 (0.045)
<i>Size</i>	0.102 (0.079)	0.154* (0.078)	0.096 (0.085)	0.156** (0.056)	0.097 (0.088)	0.153** (0.068)	0.588 (0.842)	0.144** (0.085)	0.200** (0.086)	0.427** (0.117)
<i>Age76–85</i>	-0.332 (0.544)	0.241 (0.364)	-0.380 (0.609)	0.474 (0.398)	-0.610 (0.704)	0.240 (0.365)	0.574 (0.771)	0.207 (0.355)	-0.686* (0.387)	0.502** (0.239)
<i>Age86–94</i>	0.781* (0.404)	0.121 (0.387)	0.804 (0.452)	-0.221 (0.358)	0.739* (0.452)	0.120 (0.387)	0.222 (0.365)	0.188 (0.392)	0.883** (0.447)	0.045 (0.422)
<i>Age95</i>	0.101 (0.474)	0.196 (0.336)	0.023 (0.526)	0.105 (0.294)	0.010 (0.538)	0.200 (0.337)	-0.133 (0.202)	0.442 (0.414)	0.030 (0.513)	0.862* (0.461)
<i>Specialization</i>	-0.671 (0.881)	0.185 (0.413)	-0.713 (0.946)	-0.928 (0.672)	-0.746 (1.001)	0.184 (0.413)	-0.986 (0.900)	-0.939 (0.722)	0.135 (0.822)	-1.926** (0.767)
<i>Anteopsystem</i>		0.527** (0.196)		0.425** (0.158)		0.526** (0.190)		0.493** (0.101)		0.506** (0.148)
<i>Anteapplication</i>	-1.294** (0.583)		-1.532** (0.669)		-1.519** (0.694)		-0.668** (0.104)		-0.752** (0.258)	
Controls: <i>Sector, Country</i>		Yes								
Beta0	-3.470* (1.914)	-5.096** (0.831)	-3.683* (1.997)	-3.114** (0.788)	-3.159 (2.339)	-5.096** (0.813)	-2.251** (0.777)	-2.054** (0.545)	-5.190** (1.317)	-8.280** (1.780)
Theta	-0.100 (0.869)	-0.437 (0.576)	0.0052 (0.994)	-0.426 (0.394)	-0.397 (0.964)	-0.438 (0.577)	0.054** (0.004)	-0.400 (0.666)	0.478 (0.701)	-1.041** (0.370)
Log likelihood	-810.145	-289.070	-822.277	-215.994	-812.097	-289.069	-801.142	-284.614	-863.295	-371.63
No. of observations	320	213	320	213	320	213	317	210	320	213

\*,\*\*Significant at \*10%, \*\*5%.

Notes. Nonlinear least squares estimation (second-step estimation; corrected standard errors in parentheses).

potential cannibalization threat that OSS products exert on proprietary software products. Firms with a large portfolio of proprietary software trademarks fear the devaluation of their brand name and reputation for quality if they straddle to OSS. Our third finding reinforces the idea that complementarities are a key factor when appropriability conditions are weakened (Teece 1986). We show that firms with large stocks of hardware trademarks would benefit from OSS because it commoditizes those portions of the value chain in which they do not have a competitive advantage. Interestingly, we do not find that such complementarities are present when we analyze the stock of hardware patents. We conjecture that the positive effect of complementarity can be offset by the presence of fundamental differences in programming languages between hardware and software engineers.

Our findings generate some important implications for managers and practitioners. First, OSS moves the core of appropriability from code secrecy to the legal protection of software technology, i.e., the protection

of an algorithm. Consequently, the technology core of a software product will be less dependent on the code writing ability that makes an algorithm executable, and more dependent on owning and protecting the algorithm that accomplishes a task best. This has important implications for firms that aim to exploit OSS opportunities because they might need important investments in patent protection as well. Most notably, small and young firms with scant legal protection experience could be placed at a disadvantage if they do not quickly realize the importance of formally protecting their core technologies.

Second, mixing open source approaches with proprietary assets and investments is the way for firms to profit from commercializing OSS. This could be important not only for large incumbents, but also for entrepreneurial firms (see Gruber and Henkel 2006). An example of this approach is Squeezebox, which, as we mentioned, mixes OSS (the SlimServer) with patented software components (some key codecs) and embeds the whole package with a proprietary hardware device. Although the literature has suggested that mixing proprietary assets

with OSS can enhance appropriability (see, for instance, Dahlander and Magnusson 2005, Henkel 2006), our work generalizes this finding to a large sample of OSS products. We acknowledge, however, that because tight control of intellectual property rights can discourage participation of independent developers (Shah 2006), this approach might not work well when OSS projects are in their initial stages and require a great deal of contribution from the community.

Finally, our results highlight some implications for the evolution of the value chain between software and hardware if OSS continues to gain importance. If quickly riding the gravy train is the straightforward conclusion for hardware vendors, our suggestion to specialized software suppliers is to shift their investments towards novel selling channels, new relationships with customers, and different marketing approaches according to the growth of OSS markets. However, our findings suggest that this change is particularly difficult for consolidated leaders in the software industry. This also confirms the serious potential threat that the OSS movement exerts on these firms.

There are many important aspects of OSS commercialization that we were unable to analyze in this paper. One interesting research question that we did not analyze for lack of data is how firms use different strategies to limit the amount of source code revealed in their OSS products. The degree of openness would be an important piece of information to elucidate how firms profit from OSS products. Nor did we have extensive information about the type of license under which the software product was sold (although we have identified a subset of products licensed under GNU-GPL). Such information could cast new light on firm business model and commercialization strategies. Another intriguing research question concerns the degree of direct participation of firm employees in that community, and the extent to which firms hire people from that community. This analysis would help to better define the interactions between firms and the OSS community of users, practitioners, and developers (Lerner and Tirole 2005). It would also be helpful to understand how firms resolve organizational challenges that arise from coordinating commercial needs with the philosophy of the OSS community, specifically, how firms coordinate and motivate OSS contributors across multiple teams to establish realistic goals and deadlines. Finally, we have tried to understand empirical patterns emerging from the analysis of a large data set of secondary data. However, to better assess the appropriateness and viability of different business models based on the commercialization of OSS products, one needs to rely on more micro and detailed information. These, we believe, are all potential avenues for future research.

## Acknowledgments

The authors thank Ashish Arora, Shane Greenstein, Isabel Gutierrez, Salvatore Torrisi, Brian Silverman, Senior Editor Olav Sorenson, three anonymous reviewers, and seminar participants at the London Business School, Boston University, London School of Economics, Copenhagen Business School, Bocconi University, IESE Business School, Instituto de Empresa, KUL University of Leuven, the 12th Product Development Management Conference (Copenhagen), the 10th DRUID Conference (Copenhagen), the 2005 Harvard Business School Strategy Research Conference, the 2006 Academy of Management Annual Meeting (Atlanta), and the 2007 Wharton Technology Conference for comments and suggestions on earlier drafts of this manuscript. Financial support from the Centre Cournot pour la Recherche en Economie (A. Luzzi), the Spanish Ministry of Science and Technology (SEC2003-03797 and SEC2005-06655), and the Fundación Ramón Areces is gratefully acknowledged. All errors are our own.

## Appendix. Search Methods and Error Testing

We use two search algorithms for patents and trademarks to construct algorithms that reduce the probability of not extracting a correct hardware or software trademark or patent (to reduce type II errors). For software patents, we search in the patent abstract for the following string of words: ["computer software" or "operating system" or "computer program" or "software algorithm" or "data processing" or "software application"]. For hardware patents, we search for the string ["computer server" or "computer hardware" or "motherboard" or "peripherals" or "workstation" or "mainframe" or "disk driver" or "area network"]. For trademarks, we apply the same algorithms (and same strings of words) to the text of the trademark description of goods and services. To validate the accuracy of the algorithms, we compared the results of our search with a random sample of 200 patents and trademarks read and classified by a software engineer into software/no software or hardware/no hardware patents and trademarks. Using a conservative approach, we selected the random sample inside the general electronic technological and product classes, excluding too-distant classes (like food, textiles, firearms, etc.). The error percentages were as follows. From 107 nonsoftware patents, the chosen algorithm spotted 23 patents (21.4%) as software patents. From 93 software patents, 8 patents (8.6%) escaped from the algorithm search. Similarly, from 84 nonsoftware trademarks, 12 (14.2%) were included as software, and from 116 software trademarks, 9 (7.7%) escaped from the algorithm search. For hardware, from 104 nonhardware patents, we had 16 errors (15.3%), and from 96 hardware patents, we had 9 errors (9.3%). From 124 nonhardware trademarks, we had 18 errors (14.5%), and from 76 hardware trademarks, we had 6 errors (7.8%). As expected, errors for hardware are less severe than for software, and errors in trademarks are less severe than for patents. Overall, the error percentages are similar to those of Bessen and Hunt (2007).

## Endnotes

<sup>1</sup>Although all types of OSS licenses are required to make available their source code, they might differ on the restrictions imposed on derivative works. At one end of the spectrum, copyleft licenses (for instance, the GNU-GPL) force

subsequent programs to be licensed under the same conditions. At the other extreme, noncopyleft licenses (for instance, the Berkley Software Distribution (BSD) license) might allow redistribution of derivative works under any license scheme because they impose no burden of reciprocity upon the licensee.<sup>2</sup> These press announcements provide a measure of firm efforts at commercializing OSS products. Although they do not allow us to distinguish between investments in simply assembling the product and investments in developing it, this distinction does not represent a serious concern for the arguments proposed in our hypotheses.

<sup>3</sup>The standard way to model heterogeneity in the Poisson model is to respecify the unconditional distribution for  $y_i$  as a negative binomial (hypothesizing the error term  $\varepsilon_i$  in the conditional Poisson distribution of  $y_i | \varepsilon_i$  to have a gamma  $(1, \xi)$  distribution). A normal error term is used here because the standard formulation does not permit addressing the missing zero problem (Terza 1998).

<sup>4</sup>Because we are estimating a count model, this means specifically that doubling the number of software patents with respect to the mean would imply a predicted change in the number of OSS product introductions from 1.89 to 2.77.

<sup>5</sup>These counterfactual increases in the core independent variables represent quite conventional means for measuring their size effect on the dependent variable according to the estimation model. They are meant to give a broad idea of the results of an experiment that might be unfeasible in reality, given the nature of firm resources (Barney 1991). It is also important to note that our independent variables show high standard deviations within our sample.

## References

- Arora, A. 1995. Licensing tacit knowledge: Intellectual property rights and the market for know-how. *Econom. Innovation New Tech.* **4** 41–59.
- Barney, J. B. 1991. Firms resources and sustained competitive advantage. *J. Management* **17** 99–120.
- Bessen, J., R. M. Hunt. 2007. An empirical look at software patents. *J. Econom. Management Strategy* **16** 157–189.
- Bonaccorsi, A., C. Rossi, S. Giannangeli. 2006. Adaptive entry strategies under dominant standards: Hybrid business models in the Open Source software industry. *Management Sci.* **52**(7) 1085–1098.
- Chaudhuri, A., M. B. Holbrook. 2001. The chain of effects from brand trust and brand affect to brand performance: The role of brand loyalty. *J. Marketing* **65**(2) 81–93.
- Cohen, W., R. Nelson, J. Walsh. 2000. Protecting their intellectual assets: Appropriability conditions and why U.S. manufacturing firms patent (or not). NBER Working Paper 7552. NBER, Cambridge, MA.
- Conner, K. R. 1998. Strategies for product cannibalism. *Strategic Management J.* **19** 9–26.
- Dahlander, L., M. G. Magnusson. 2005. Relationships between open source software companies and communities: Observations from Nordic firms. *Res. Policy* **34**(4) 481–493.
- Dam, K. W. 1995. Some economic considerations in the intellectual property protection of software. *J. Legal Stud.* **24**(2) 321–377.
- Dosi, G. 1988. Sources, procedures and microeconomic effects of innovation. *J. Econom. Literature* **26**(3) 1120–1171.
- Economist, The.* 2005. An open secret. (Oct. 20) 31–33.
- eWeek.* 2005. Novell's earnings fall short. (Aug. 25) 8.
- Gambardella, A., B. Hall. 2006. Proprietary vs. public domain licensing of software and research products. *Res. Policy* **35**(6) 875–892.
- Gandal, N., S. Greenstein, D. Salant. 1999. Adoptions and orphans in the early microcomputer market. *J. Indust. Econom.* **47**(1) 87–105.
- Garud, R., A. Kumaraswamy. 1993. Changing competitive dynamics in network industries: An exploration of Sun Microsystems' open systems strategy. *Strategic Management J.* **14**(5) 351–369.
- Gatignon, H., M. L. Tushman, W. Smith, P. Anderson. 2002. A structural approach to assessing innovation: Construct development of innovation locus, type, and characteristics. *Management Sci.* **48**(9) 1103–1122.
- Ghemawat, P. 1991. Market incumbency and technological inertia. *Marketing Sci.* **10**(2) 161–173.
- Green, W. H. 2000. *Econometric Analysis*, 5th ed. Prentice Hall, Upper Saddle River, NJ.
- Gruber, M., J. Henkel. 2006. New ventures based on open innovation: An empirical analysis of start-up firms in embedded Linux. *Internat. J. Tech. Management* **33**(4) 256–372.
- Hall, B., M. MacGarvie. 2006. The private value of software patents. NBER Working Paper 12195. NBER, Cambridge, MA.
- Hall, B., R. Ziedonis. 2001. The determinants of patenting in the U.S. semiconductor industry, 1980–1994. *RAND J. Econom.* **32** 101–128.
- Hardin, J., J. Hilbe. 2001. *Generalized Linear Models and Extensions*. Stata Press, College Station, TX.
- Hecker, F. 1999. Setting up shop: The business of open source software. *IEEE Software* **16**(1) 45–51.
- Henkel, J. 2006. Selective revealing in open innovation processes: The case of embedded Linux. *Res. Policy* **35** 953–969.
- King, A. A., C. L. Tucci. 2002. Incumbent entry into new market niches: The role of experience and managerial choice in the creation of dynamic capabilities. *Management Sci.* **48**(2) 171–186.
- Koenig, J. 2004. Seven open source business strategies for competitive advantage. *IT Manager's J.* (May 14).
- Lerner, J., J. Tirole. 2002. Some simple economics of open source. *J. Indust. Econom.* **50**(2) 197–234.
- Lerner, J., J. Tirole. 2005. The scope of open source licensing. *J. Law, Econom., Organ.* **21** 20–56.
- Mann, R. J. 2006. Commercializing open source software: Do property rights still matter? *Harvard J. Law Technology* **20** 1–46.
- Mitchell, W., K. Singh. 1992. Incumbents' use of preentry alliances before expansion into new technical subfields of an industry. *J. Econom. Behav. Organ.* **18**(3) 347–372.
- Murphy, K. M., R. H. Topel. 1985. Estimation and inference in 2-step econometric models. *J. Bus. Econom. Statist.* **3**(4) 370–379.
- O'Mahony, S. 2003. Guarding the commons: How community managed software projects protect their work. *Res. Policy* **32**(7) 1179–1198.
- O'Mahony, S., F. Ferraro. 2007. The emergence of governance in an open source community. *Acad. Management J.* **50**(5) 1079–1106.
- Personal Computer World.* 2006. Let the Linux music play. **1** 15.
- PR Newswire.* 2003. Microsoft blocks distribution of desktop Linux in Sweden; Judge issues temporary restraining order halting LinuxOS sales to Swedes. **7**(12, Dec. 11).
- Roberts, J., I. Hann, S. Slaughter. 2006. Understanding the motivations, participation and performance of open source software developers: A longitudinal study of the Apache Projects. *Management Sci.* **52**(7) 984–999.

- Samuelson, P. 2006. IBM's pragmatic embrace of open source. *Comm. ACM* **49**(10).
- Seethamraju, C. 2003. The value relevance of trademarks. J. R. M. Hand, B. Lev, eds. *Intangible Assets: Values, Measures and Risks*. Oxford University Press, Oxford, UK.
- Shah, S. K. 2006. Motivation, governance and the viability of hybrid forms in open source software development. *Management Sci.* **52**(7) 1000–1014.
- Smith, G. V., L. R. Parr. 2000. *Valuation of Intellectual Property and Intangible Assets*, 3rd ed. John Wiley & Sons, New York.
- Spaeth, S., M. Stuermer, S. Haefliger, G. von Krogh. 2006. Sampling in open source software development: The case for using the Debian GNU/Linux distribution. Unpublished manuscript, ETH Zurich, Switzerland.
- Teece, D. 1986. Profiting from technological innovation: Implications for integration, licensing and public policy. *Res. Policy* **15**(1) 285–305.
- Terza, J. V. 1998. Estimating count data models with endogenous switching: Sample selection and endogenous treatment effects. *J. Econometrics* **8** 129–154.
- Torrise, S. 1998. *Industrial Organisation and Innovation: An International Study of the Software Industry*. Edward Elgar, Cheltenham, UK.
- Tripsas, M. 1997. Unraveling the process of creative destruction: Complementary assets and incumbent survival in the typesetter industry. *Strategic Management J.* **18** 119–142.
- von Hippel, E., G. von Krogh. 2003. Open source software and the “private-collective” innovation model: Issues for organization science. *Organ. Sci.* **14**(2) 209–223.
- von Krogh, G., S. Spaeth, S. Haefliger. 2005. Knowledge reuse in open source software: An exploratory study of 15 open source projects. *Proc. 38th Hawaii Internat. Conf. System Sci.*, 198–207.
- Ziedonis, R. H. 2004. Don't fence me in: Fragmented markets for technology and the patent acquisition strategies of firms. *Management Sci.* **50**(6) 804–820.