To be published in the *Proceedings of the Hawai‘i International Conference on System Sciences* (HICSS-40), Waikola, Hawai‘i, January 3-7, 2007. © 2007 IEEE.

Value Capture and Value Networks in Open Source Vendor Strategies

Joel West
San José State University
Joel.West@sjsu.edu

Abstract

Firms involved in open source software face inherent limits on their ability to appropriate returns from technological innovation. Here I examine the business models used by IT vendors given the limited appropriability available for open source software. I show how firms capture value through complementary assets, while creating value and positive network effect through an inherent openness that attracts complementors, users and rivals to their corresponding value network. From this, I offer suggestions for future research on complementary assets and value capture in business models.

1. Introduction

Prior research has identified three fundamental aspects of a business model: creating value, capturing value, and embedding the business model into a value network. Much of the managerial and academic interest in business models has focused on how to appropriate the value created from new Internet-enabled businesses.

A major category of Internet-enabled business models relates to the use of open source software. A key issue for such models is how firms capture value without formal appropriability of an innovation (West, 2003). The interest in open source business models has existed nearly as long as the term “open source” itself, centering on the apparent paradox of firms making money from “free” software.

After reviewing prior research related to business models, this paper presents the findings of a four-year field study of open source business models. It identifies two basic modes for such models, in-licensing and out-licensing. It also shows the role that complementary assets play in value capture, particularly in those cases where firms lack formal appropriability for the open source code. Finally, the paper suggests how the assets for information goods are both similar to and different from other research on complementary assets.

2. Complements and Business Models

2.1 Complementary Assets

While firms can appropriate value created from technological innovation through formal mechanisms such as intellectual property laws, such strong appropriability is rare. Instead complementary assets play an important role allocating the returns for such innovation (Teece, 1986; West, 2006). Such assets also help firms cope with later changes that render existing technologies obsolete (Tripsas, 1997; Rothaermel, 2001). For many innovations, complementary goods or services are essential for completing the “total” or “whole” product demanded by buyers (Teece, 1986; Moore, 1991; Brandenburger and Nalebuff 1996).

A firm’s relationship to the network of providers of complementary products determines its value creation, value capture and the durability of its competitive advantage; thus creating and managing these relationships is an important part of achieving each of these goals (Hamel, 2000; Iansiti and Levien, 2004). The innovator must incent third party suppliers of these complementary assets to complete the innovation, particularly when the complementor has to make investments that are specialized to the particular innovation (Teece, 1986).

2.2 Business Models

In prior research, the two most commonly identified dimensions of an innovation business model are creating and capturing value from that innovation. Both aspects are often mediated by the firm’s relationships with other firms in a value network — firms that supply inputs, buy outputs and provide complementary goods and services (Table 1).

Value Creation. Not surprisingly, value creation is a universal dimension of recent conceptions of a business model (Magretta, 2002; Chesbrough & Rosenbloom, 2002; Morris et al., 2005; Shafer et al., 2005). Given heterogeneous preferences across both individual and organizational buyers, value creation requires identifying relevant customer segments, the value proposition for each, and how the business model will provide that value (Chesbrough and Rosenbloom, 2002; Morris et al., 2005).

Sustainable Value Capture. A business model must also explain how a firm captures value from its value creation. One key step is to define a revenue model, including what will be priced and how much will be charged (Amit and Zott, 2001; Chesbrough and Rosenbloom, 2002). Another prerequisite is making sure that the cost structure is consistent with the customer's perceived value and the portion that can be captured (Margetta, 2002; Chesbrough and Rosenbloom, 2002). Finally, firms must sustain that value capture against competition, often through unique and valuable internal characteristics such as resources, capabilities, core competencies, or culture. Firms can also establish durable external relationships with the buyers, sellers and complementors within a value network (Oliver, 1997; Stabell and Fjelstad, 1998; Iansiti and Levien, 2004).

Value Network. Because the value created and captured by a firm is determined by its position in the value chain, a firm's business model must define its role in the value chain (Amit and Zott, 2001; Chesbrough and Rosenbloom, 2002; Margetta, 2002). Many firms also depend on the supply of third-party complements, and thus the firm's strategy depends on the entire value network of suppliers, buyers, complementors and other allies (Amit & Zott 2001; Chesbrough and Rosenbloom, 2002; Afuah, 2000).

These three dimensions of a business model are implemented through a firm's strategies and its execution of those strategies. For the purpose of our study, consistent with Magretta (2002) I adopt the definition that a business model is a set of strategic decisions and assumptions that are rarely (if ever) revisited by firms for a given business activity.

3. Proprietary and Open Source

This study looks at the business model of information technology (IT) firms that utilize open source software (OSS). Some are incumbent IT producers, while others are *de novo* OSS-focused startup companies. For some, the primary revenues come from software, while others depend on selling hardware, support or other goods and services.

3.1 Production of Software Goods

Software company business models have generally focused on services, standardized package software, or some combination thereof (Campbell-Kelly, 2003; Cusumano, 2004). Software products may be sold as a standalone good, or as part of an overall system (such as a PC or a network router).

Software companies spend a relatively high proportion on research and development: the NSF (1999) put U.S. package software R&D intensity at 15.0%, versus 5.8% for computers and 2.8% for machinery and electrical equipment. With large up front R&D and low duplication costs, package software business models enjoy tremendous economies of scale. Such models depend heavily on intellectual property protection (e.g. copyright and trade secrets) that provides only limited protection against imitation (Davis et al, 1996).

However, only a fraction of this R&D spending goes for new-to-the-world innovations. Instead, the development of software tends to be incremental, cumulative and more analogous to industrial design than scientific invention. Because source code is a textual representation of a computational machine, this code can easily be adapted and recombined to solve new problems (Samuelson et al, 1994).

Despite software's inherent flexibility and modularity, such recombinations tend to happen within the boundary of a firm, as the disclosure of source code raises serious risks of misappropriation and thus impairs the creation of markets for software components. As a result, annual expenditures on re-implementing existing software systems have been estimated in the billions of dollars (Teece, 1998).

3.2 Free and Open Source Software

"Open source" and "free software" have their origins in university software research projects during the 1980s. The movements share principles related to source code availability, but differ primarily in free software's insistence on compulsory sharing of derivatives of such code through the "copyleft" requirement (Rosen, 2005).

Interest in open source has focused on well known packages such as Linux and Apache, developed by self-organized voluntary communities. The activity both between and within such projects is highly skewed towards a small number of major projects and major contributors within each project (Healy & Schussman 2003; Crowston et al 2006). More recently, firms have sponsored their own open source projects, retaining control and influence not seen in autonomous projects (West and O'Mahony, 2005).

Some IT vendors have incorporated OSS as part of their business models. Large firms such as IBM and Apple have leveraged community open source products to replace portions of their software architectures (West, 2003). In other cases, new firms have created new business models around open source of their own creation (Välimäki, 2003, 2005). The most widely supported open source software has been the web server architecture dubbed "LAMP," which consists of the Linux operating system, Apache web server, MySQL relational database and three web scripting languages (Perl, Python and PHP). Of these, multiple firms focus on distributing Linux, while the original authors of both MySQL and PHP created venture capital-funded startup firms to profit from their respective open source technologies.

3.3 Research Design

Consistent with accepted principles of inductive theory generation from qualitative data (Eisenhardt, 1989; Yin, 1994), this study of open source business models is based on qualitative data from a theoretical sample of firms involved in open source software. From 2002-2005 I interviewed 70 informants representing 44 organizations, including software companies, other vendors of I.T. products and services, organizations considering adoption of OSS, and representatives of key open source projects. Of the 70 informants, 83% were US-based informants working for US-headquartered organizations.

The semi-structured interviews typically were conducted in person and lasted from 45 to 90 minutes. As recommended by Glaser and Strauss (1967), site selection and protocol evolved over time. The interview data (contemporaneous notes and the recordings or transcripts) were supplemented by press coverage of open source topics. The data was then reviewed to identify convergent and divergent interpretations, both within informant role (vendor, user, OSS contributor) and between different roles.

The initial research question was "How do firms capture value without formal appropriability?" As the data collection progressed, it was refined to concentrate on the complementary relationship between the priced and unpriced components of IT vendor products and services.

The findings of this paper look at how IT vendors address each of the three dimensions of business models with open source, and the strategies that are derived from such models.

4. Creating Value

4.1 Buyer Perceptions

In interviews, buyers identified two ways that OSS created value compared to proprietary software:

Lower prices. Successful OSS packages (both sponsored and community) enjoyed strong word of mouth recognition and distribution. This free marketing, in turn, increased efficiency for key open source companies. The CEO of SugarCRM explained

> Every dollar you give [competitor], 70 cents to goes to fund the sales and marketing efforts, and maybe 11-14% actually goes to pay the engineering salaries that write the code (interview, John Roberts, June 6, 2005)

Packages were also cheaper to use if they built upon cheap complements. During the late 1990s, the "LAMP" stack was developed and deployed on inexpensive personal computers rather than Unix-based systems, thus reducing hardware costs.

Less lock-in. Historically, vendors had used high switching costs as a way to lock in buyers and extract rents (Shapiro and Varian, 1999). The availability of source code and lower barriers to imitation inherently lowered vendor power, as with the earlier open systems movement (West, 2003).[1]

Despite this indirect benefit, source code had only limited direct utility. Free software advocates argue that the value of free software means "free as speech, not free as in beer." But organizational adopters clearly preferred not to make changes to code — except to cure urgent defects, or to add capabilities to newer projects that lacked a mature feature set.

Of course, a few packages were popular because of specific feature or performance advantages. The Apache web server offered better standards compliance and integration with other technologies than its less popular rivals, while MySQL used less computer resources than its proprietary competitors.

4.2 Priming the Positive-Feedback Loop

The initial adopters of an OSS package help prime the pump of the positive-feedback adoption loop variously described as "demand side economies of scale" (Katz & Shapiro 1986) or "increasing returns to scale" (Arthur, 1996). As with packaged software, both the producer and adopters of OSS benefit from the positive network effects that arise from the increased supply of third party complements. The choice of an OSS business model thus reflects a belief that weak value capture for a widely adopted technology provides greater returns than strong value capture for a sparsely adopted technology.

In a network effects environment, a major objective of technology adoption is to foreclose any possible competitor (Arthur, 1996; Shapiro and Varian, 1999). For example, both IBM (with Eclipse) and Real Networks (with Helix) sponsored open source projects hoping to create winning platforms in competition with Microsoft's proprietary alternatives.

By its nature, OSS appeals to particular users and enable specific distribution opportunities. By conforming to the license terms of the Open Source Initiative, OSS packages (whether sponsored or community-controlled) are incorporated into — or distributed with — other open source packages, such as Linux, Apache and Eclipse. As the CEO of Sleepycat Software explained:

> Open Source distribution gives you a ubiquity that you just can't buy for any amount of marketing dollars. (Interview, Michael Olson, Feb. 1, 2005).

By attracting a large pool of adopters, firms also gain a large pool of potential contributors to increase

[1] While buyers said they valued the opportunity to switch vendors (e.g. between SuSE and Red Hat Linux), we did not find an example where that where such a switch had occurred. Firms appeared to value an option they were unlikely to exercise.

value of their software. These contributions may come as enhancements to the core OSS technology (unique to OSS) or through development of separate complements (which is not).

4.3 Market Segmentation

Value creation is only relevant in the context of a specific segment of potential buyers, which for open source have largely been business buyers.

One possible explanation is the path dependence of early open source development, which focused on infrastructure technologies around a Unix clone and Internet infrastructure. Much of the early success of open source came in the booming area of Internet servers, which were of interest to firms providing e-commerce, company web pages and intranets. Not coincidentally, the most widely adopted consumer open source project is the Mozilla browser, a crucial technology supported by systems vendors that needed Internet applications to sell computers.

Also, open source has been presumed to be most attractive to businesses (as well as universities and hobbyist-programmers), who have the technical skills and resources to utilize cheap unsupported software. They are also the most likely to contribute changes to support an OSS project's development.

However, instead of "free beer," industry analysts now talk about a "free puppy," with no initial cost but ongoing upkeep costs.[2] Though large business buyers have the ability to support the software on their own, interviews showed that they are both willing to outsource peripheral activities and are also the most risk averse when it comes to critical computer systems. Overall, business buyers of computer systems seem to expect a richer whole product solution — integration, customization, integration, support and upgrades — which create more opportunity for vendors to combine priced and unpriced complementary assets that create value.

4.4 Commoditization

Some OSS packages offer new-to-the-world innovations. For example, both Apache and Mozilla are open source descendants of the early web server and browser developed at the National Center for Supercomputing Applications in the early 1990s. But other popular OSS packages are merely re-inventions of existing software to create an open source alternative — whether Linux, Project GNU, Berkeley DB or MySQL.

In many cases, the use of OSS is a tacit (or explicit) commoditization of a previously proprietary technology. If the software provides few options to differentiate through additional features, then firms may increase the customer's perception of value added by reducing the cost of producing an undifferentiated product (Porter, 1985: 3).

Such cost reduction is consistent with those from the widespread adoption of personal computers. IBM's modular hardware architecture — and the division of labor between the various suppliers of that architecture — brought componentization, experimentation, and improvement in both performance and cost. The supplier of dominant components enjoyed unmatched scale efficiencies (Grove 1996).

A popular OSS package not only enjoys the network effects identified by Grove, but, unlike proprietary technologies, makes it attractive for a competing development efforts (whether proprietary or OSS) to adopt the free software. This software sharing enables cumulative innovation and breaks the re-invention drag identified earlier by Teece (1998).

5. Capturing Value

OSS has lower formal appropriability than with proprietary software. The source code is developed in a highly visible, virtual community enabled by Internet infrastructure That code is available for reuse and modification by competitors, complementors and customers under an approved open source license, thus providing the vendor less appropriability than if it used proprietary code covered under trade secrets.

Firms that own the code in an OSS project enjoy some limited measure of formal appropriability by excluding rivals and selling key rights to users. But because the firm cannot directly appropriate all the returns from the code, even then its business model depends on selling complementary goods or services to capture value, or leveraging tacit knowledge or other intangible source of advantage over rivals.

5.1 Differences in Formal Appropriability

Discussions with vendors about their open source strategies drew a clear demarcation between two fundamentally different open source strategies: those where the firm controls the software's source code and those where it does not. — which Rosen (2005) refers as "out-licensing" and "in-licensing."

The choice is not entirely endogenous, in that both depend on a relevant supply of code. For out-licensing, a firm most both have its own code to disclose, and be willing to cannibalize revenues for that code by turning it into a commodity. Out-licensing can be further subdivided into two cases: whether the firm maintains or surrenders ownership.

[2] The earliest published references attribute the "free puppy" metaphor to executives from Microsoft (Salkever 2002) and Sun (Krill 2002), whose proprietary operating systems both compete with Linux.

Conversely, the in-licensing mode depends on the existence of OSS code available outside the firm, whether created by the community or through the out-licensing of a competitor or complementors. It also presumes the code is offered under terms that fit with the in-licensor's planned business model.

5.1.1 Ongoing Ownership

When a firm releases its code as open source, it voluntarily waives trade secret protection. If it retains the copyright, it can thus specify the terms under which others may license the code (Rosen, 2005).

An example of such firm-owned code is the "dual license" approach employed by MySQL AB and Sleepycat Software. Such software is available free under the copyleft restrictions of an approved license (usually the Free Software Foundation's GPL). To avoid these restrictions, user firms can pay to license the software under less restrictive terms (Välimäki, 2003, 2005). The dual license is a straightforward application of price discrimination consistent with Shapiro and Varian (1999). Dual license firms maximize yield by tailoring the two licensing options to the differences between the price-sensitive and price-insensitive user populations (Välimäki, 2005).

The licensing obligations are usually acceptable to individual and non-commercial users, but are unacceptable to two classes of commercial entities. Some organizations incorporate the GPL-licensed code in their own internal systems, and are wary of disclosing their systems to rivals. In other cases, vendors incorporate the code as part of their own "whole product" solution, such as vertical market systems vendors or those design computer software to embedded in tangible products such as routers. The CEO of MySQL AB emphasized the company's goal of capturing some of the latter revenue:

> We don't mind people using our product free of charge, but we do mind if people make money on our product without us getting a share. (Interview with Mårten Mickos, Sept. 17, 2003).

Firms that out-license retain ongoing control through control of their IP rights. Their release of software to outside parties under restrictive terms allows them to make it unattractive for rivals to use (West, 2003). To maintain this control, firms usually require a full or partial assignment of IP rights for any code contributed by users; in extreme cases, firms rewrite all contributions so that they own the IP (Välimäki, 2003).

A few companies maintain ownership of code, but release it under so-called "permissive" licenses (like the BSD or Apache license) which allow rivals or users to combine the code without restrictions. Other firms use a "weak" copyleft for OSS components, which compels disclosure of changes to that component but (unlike the GPL) not changes to modules built using that component.

5.1.2 Surrendering Ownership

With out-licensing, firms may waive their existing appropriability claims by assigning their ownership (copyright) and/or development control to an external body. Firms may also spinout their existing code to form a new community, with or without ongoing sponsorship (West & O'Mahony, 2005).

Perhaps the biggest such example is IBM's 2001 spinout of its Java-based developer tools to form the Eclipse Consortium, later formalized as a non-profit foundation. IBM donated its IP to the foundation, while it continues to provide a plurality of the ongoing development resources. However, over time new members diluted IBM's effective control over development efforts by providing both developers and their own code. Similar firm-created spinouts have also occurred within the Apache web server project, which (unlike Linux) requires code assignment to the project.

Interviews confirmed that firms surrendered formal appropriability in hopes of fueling adoption, support from key complementors and (frequently) direct competitors. The donor firm continues to use the technology as part of its existing value creation strategies, while providing development resources so the code supports its value creation and revenue mode; in other cases, the spinout provides a new home for code of no ongoing strategic value.

However, without significant external contributions, firms may actually find their costs are *increased* due to coordination costs after a spinout. One firm termed it an "Open Source" tax, estimating that it cost an extra 15-25% in developer-hours to do code development in a public context, including explaining planned changes to external stakeholders and addressing their questions and concerns.

5.1.3 External Ownership

The best known category of open source projects are the community-managed ones, as exemplified by the success of the Linux operating system founded in 1991 by Linus Torvalds. Torvalds today controls the Linux trademark, while for historical reasons the copyright to the code is held by various contributors, including the Free Software Foundation. In other cases, the communities establish formal non-profit organizations to own the copyright and other assets of the project (O'Mahony, 2003).

The Linux ecosystem illustrates the range of firm strategies to profit from community software. While firms typically contribute development resources, most of the R&D is shared across the ecosystem, thus lowering barriers to entry: rather than a single global winner, various Linux distributors have dominant market share in their home regions. Vendors of complements treat Linux as just another operating system: computer systems vendors use Linux as a replacement for Unix, while providers of software, peripherals and services make their products and services available for Linux as demand warrants.

The costs to companies vary by project and firm. In some cases, a firm provides the plurality of resources to create new technologies, as IBM has done for projects within the Apache community. In other cases, firm involvement is limited to making existing code work well with its own complementary products, as when systems vendors adapted Mozilla to work with their own systems.

5.2 Selling Complements

Except for price discrimination, all open source business models in this sample rely on selling some sort of complement. The revenue models are feasible because most OSS developers have neither the inclination nor the resources to develop the whole product solution that computer users have come to expect for commercial software. For example, installation and support services have little value to the skilled developer-contributors, but are essential for many business and individual users. Likewise, while a dedicated programmer might give away code to thousands or millions of users, he or she would be unlikely to offer direct support to all of them.

Firms' open source business models thus center on the sale of complementary products and services to complete the whole product solution, often parallel to those for proprietary software. For example, IBM will sell server hardware to run with Linux or its own operating systems; IBM's Global Services division will offer design and installation services for its own software, software from other vendors such as Oracle or SAP, OSS, or any combination thereof.

Most often, vendors use their OSS development expertise to provide various support services. In some case, firms offer a separate product that either enhances or complements the open source version, as with Zend's high-performance transaction engine for its commercial version of PHP, or the spell checker provided by Sun with StarOffice.

Advocates claim OSS users are less dependent on vendors for support services because in principle they can support themselves. But large organizations in particular said they required from OSS the same level of reliability and accountability (such as 24 hour tech support) as with their other information systems. Because most of the software provided by Linux distributors comes from publicly available sources, firms could make their own compilations and escape the per-computer support fees; however most were wary of installing a non-standard configuration and thus being without external sources of support.

5.3 Other Value Capture Strategies

Firms have two other ways of gaining advantage over rivals in capturing value from the shared software, particularly where (as with in-licensing or transferred ownership) they lack formal control.

One way is through creating and guiding the formal architecture of the OSS project. Particularly for platform technologies (such as Linux or Apache), control of the interfaces controls the supply of complementary assets (West & Dedrick, 2000). Therefore, a firm that out-licenses code (even if it waives further ownership) creates a shared platform that fits best its own internal architecture and suite of complementary products. For example, the code IBM donated to Eclipse was compatible with its own tools, while other vendors who in-licensed the Eclipse code often required several programmer-years to rewrite their code to interface with Eclipse.

Similarly, ongoing leadership of an in-licensed OSS project allows a firm's developers to influence the code to be most compatible with a firm's own requirements and software architecture, even in independent projects such as Eclipse or Apache. The issue is exacerbated with Linux, in which only a handful of individuals have the right to "commit" changes to the kernel code, so that firms without such a committer have little recourse to influence the direction of the project.

Even without influence, firms that are actively involved in the OSS development may gain technical knowledge (suitable for deployment and support) or an expert reputation useful in marketing.

5.4 Limits to Value Capture

Reduced appropriability (and thus value capture) is a fundamental difference between OSS and proprietary software business models. Even in the case where firms retain ownership of their OSS, the presence of the free version constrains pricing power.

Firms must worry about whether buyers will value the remaining elements of the whole product solution. If the complements are limited value — or there is a limited pool of buyers who value the complements — then vendors face difficulty gaining significant revenues no matter how much value the OSS creates for buyers. In fact, firms face tradeoffs between making the free version too valuable (undercutting sales of the enhanced version) or not valuable enough (slowing adoption). Most of the firms interviewed had adjusted the licensing or packaging (i.e. the division between core and complements) for their OSS.

The firms that continued to own their open source had high barriers to imitation, although their ongoing control potentially raised buyer fears about lock-in and rent-seeking. Conversely, the independent OSS project can be shared by multiple firms, reducing or eliminating the role of the core OSS code as a barrier to new or existing rivals. Such software is inherently a commodity, and thus firms faced an ongoing challenge of providing complementary products and services that are both valued by buyers and differentiated from rival complements.

6. Role of the Value Network

The magnitude of value creation from an OSS value network depends on two factors. One is the number of adopters, which attracts suppliers of complementary goods and services. The second is the proportion of adopters (typically individual or corporate programmers) who are both qualified to contribute core or complementary technologies and are willing to do so.

6.1 User Contributions

By winning a large pool of adopters, firms also win a large pool of potential contributors to increase value of their software.

Users voluntarily disclose their enhancements because they may not be able to appropriate value from minor improvements, or because they gain recognition or other intrinsic benefits from the disclosure (von Hippel, 2005). One major category of enhancements is error correction — in which users obtain direct utility from identifying and correcting their own problems. As MySQL AB's CEO bragged: "What is the cost item in our income statement for bug fixing? It's ridiculously low" (interview, Mårten Mickos, Sept. 17, 2003). Users also contribute foreign language translations of the software — as mentioned by Zend (with PHP) and Sun Microsystems (OpenOffice) — or adapt it to run on lower-priority computer systems (as with Apple's QuickTime software).

Such contributions are not without their dilemmas. At one extreme, contributors may have greater apprehension about donating code to sponsored communities, given the firm's control of the copyright and future direction (West and O'Mahony, 2005). At the other extreme, for products with a millions of users, these potential contributions may not be worth the additional effort of monitoring and responding a high volume of external suggestions.

6.2 Complementors

As with proprietary software, widespread diffusion of the software is important for attracting a supply of third-party complements. These may be complements that are integrated and sold as a whole product incorporating the core OSS package, increasing distribution. For dual-licensed OSS, these bundles also generate royalty income, as with the many products built upon MySQL or Sleepycat's Berkeley DB. Or they may be complements that are sold (or given away) separately, increasing the value of the core innovation, as with the dozens of projects created at SugarForge.org, an open source website hosted by OSS vendor SugarCRM.

At the same time, the supply of complements can compete with the OSS firm's business model. For example, Red Hat Linux has the largest market share among Linux distributions, but it competes for some service revenues with large IT service vendors such as IBM Global Services or Accenture.

6.3 Competitors

Brandenburger and Nalebuff (1996) identified four aspects of a firm's value network: suppliers, customers, complementors and competitors. Subsequent research has emphasized the role of the first three (e.g. Afuah, 2000), but here I highlight the importance of direct competitors in building legitimacy, demand and a supply of technology. Particularly in the in-licensing case, direct competitors often collaborate together to further develop or spur the adoption of their shared technology, as with the Eclipse Foundation or the Open Source Development Labs.

6.4 Zero-Sum Cost Pressures

The commoditization of software both fuels adoption and sets off a zero-sum scramble among firms to protect their value capture strategies. In other industries, this tension is between buyers and sellers in a value chain, but in the IT industry it also includes vendors of mutually complementary products.

In particular, firms talked about the ongoing trend towards "commoditizing up the stack" (Figure 1). The highest levels of the stack are most visible to buyers, and thus firms at the top levels hoped to commoditize the lower levels. One proprietary software vendor said their efforts to commoditize lower layers were part of a conscious effort to maximize "wallet share": as buyers seek to cut IT budgets, reducing the cost of lower layers allowed it to claim a bigger share of the buyer's spending.

Conversely, the developers of each layer — whether proprietary, out-licensed OSS or in-licensed OSS — sought a positive-sum outcome, by attracting higher-level third party complements that make their offerings more attractive. As predicted by Teece (1986), complement suppliers that must make a specialization investment to support a specific lower layer are reluctant to support too many technologies. Thus, while application vendors wrote their software for "Linux," most offered support for specific distributions such as "Red Hat Enterprise Linux" or "SuSE Linux Enterprise Server." Buyers that relied on such complements constrained their consideration set to those distributions certified by application vendors — thus reducing the theoretically high level of competition among Linux distributions to a *de facto* duopoly. Buyers that used Linux to run their own applications faced no such constraints, because their own code could be modified to work with whatever version they selected.

A key way for a firm to attract complements was to provide its software to complement providers on attractive terms, as when MySQL and Sleepycat's Berkeley DB are given away with leading Linux distributions as well as on their own websites.

7. Discussion

In most cases, the OSS revenue model depends on the sale of complementary goods and services necessary to complete the whole product expected by buyers such as large organizations. Although the core software technology would be expensive to replicate, many of the OSS packages represent mature technologies that became commodities through their release as open source.

While capturing value is inherently more difficult, these OSS models offers three interrelated advantages when compared to conventional models: a credible commitment to prevent underinvestment in complementary assets within the value chain, priming the positive-returns network effects cycle, and various efficiency and scale economies.

The terms of an open source license represent a credible commitment that both the terms and price for use of the technology are granted in perpetuity. This means that both user-adopters and third party complementors can invest in specializing their complementary assets for the technology without fear of rent seeking. Openness also incents user adopters to reveal internal complements for use by others.

This initial nucleus of users and complements primes the cycle of positive network effects. Given such effects, firms that reduce their appropriability by out-licensing their technology favor widespread adoption (with value capture difficulties) over strong value capture at the risk of sparse adoption.

When it achieves widespread adoption, an OSS business model also benefits from key efficiencies and scale economies in error correction, manufacturing, marketing, and distribution.

Such dynamics have the potential to more broadly change the software industry. As Teece (1998) notes, component markets for software are impaired by the failure of firms to reliably capture a return on their software due to imperfect appropriability. If firms don't need to directly sell software — but instead capture value through indirect revenue models — then this would both enable cumulative innovation and reduce duplicative software development costs.

There are limits to generalizing from this study. The interviews were intended to be representative of open source business strategies, but any non-random sample is subject to sampling bias. In addition, the sampling frame was disproportionately oriented towards U.S.-based firms, although other countries have more favorable attitudes towards OSS.

Future research could establish the generality of the findings regarding waiving appropriability and value capture through complementary assets. Prior research has emphasized the use of external suppliers of complementary assets in cases where they are not critical or the firm is unable to produce them. The decision by large multinationals to switch from their own technologies to external complements does not fit this pattern, but is it particularistic to IT industry or a viable strategy for other industries as well?

Also, earlier research on complementary assets has emphasized their role in profiting from radical technological innovation. This study has shown their value in cases of mature established technologies — where the primary innovation is experimentation in business models rather than technologies or processes — but other industry contexts should be studied.

8. References

Afuah, A. 2000. "How much do your co-opetitors' capabilities matter in the face of technological change?" *Strategic Management Journal* 21 (3), 397 - 404.

Amit, R., Zott, C. 2001. "Value creation in e-business." *Strategic Management Journal* 22 (6-7), 493-520.

Arthur, W. B. 1996. "Increasing Returns and the New World of Business," *Harvard Business Review,* 74 (4), 100-109.

Brandenburger, A., Nalebuff, B.J. 1996. *Co-opetition.* Doubleday, New York.

Campbell-Kelly, M, 2003. *From airline reservations to Sonic the Hedgehog: a history of the software industry.* MIT Press, Cambridge, Mass.

Chesbrough, H. and Rosenbloom, R. S. 2002. "The role of the business model in capturing value form innovation: evidence from Xerox corporation's technology spin-off companies." *Industrial and Corporate Change,* 11 (3), 529-555.

Crowston, K., Wei, K., Li, Q., Howison, J. 2006. "Core and periphery in Free/Libre and Open Source software team communications." *Proceedings of the 39th Hawaii International Conference on System Sciences.*

Cusumano, M. A. 2004. *The Business of Software*. Free Press, New York.

Davis, R., Samuelson, P., Kapor, M., Reichman, J. 1996. "A new view of intellectual property and software." *Communications of the ACM,* 39 (3), 21-30.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research." *Academy of Management Review* 14 (4), 532-550.

Glaser, B.G., Strauss, A. 1967. *The Discovery of Grounded Theory: Strategies of Qualitative Research.* Wiedenfeld and Nicholson, London.

Grove, A. S. 1996. *Only the paranoid survive: how to exploit the crisis points that challenge every company and career*. Currency Doubleday, New York.

Hamel, G. 2000. *Leading the Revolution*. Harvard Business School Press, Boston.

Healy, K., Schussman, A. 2003. "The Ecology of Open-Source Software Development". working paper, Department of Sociology, University of Arizona.

Iansiti, M., Levien, R. 2004. *The Keystone Advantage: What The New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability.* Harvard Business School Press, Boston.

Katz, M. L., Shapiro, C. 1986. "Technology Adoption in the Presence of Network Externalities." *Journal of Political Economy* 94 (4), 822-841.

Krill, P. 2002. "Sun ponders offering open-source Solaris; Exec questions effectiveness of concept" *InfoWorld.com,* Dec. 5, URL: http://www.infoworld.com/articles/hn/xml/02/12/05/021205hnopensolaris.html

Magretta, J. 2002. "Why business models matter." *Harvard Business Review* 80(5), 86-92.

Moore, G. A. 1991. *Crossing The Chasm: Marketing and Selling Technology Products To Mainstream Customers,* HarperBusiness, New York.

Morris, M., Schindehutte, M., Allen, J. 2005. "The entrepreneur's business model: toward a unified perspective." *Journal of Business Research* 58 (6), 726-735.

NSF. 1999. "US Corporate R&D, Volume I: Top 500 Firms in R&D by Industry Category." National Science Foundation, Washington, D.C.

O'Mahony, S. 2003. "Guarding the Commons: How Community Managed Software Projects Protect Their Work." *Research Policy* 32 (7), 1179-1198.

Oliver, C. 1997. "Sustainable Competitive Advantage: Combining Institutional and Resource-Based Views." *Strategic Management Journal* 18 (9), 697-713.

Porter, M. E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*, Free Press, New York.

Rosen, L. 2005. *Open Source Licensing. Software Freedom and Intellectual Property Law*. Prentice Hall PTR, Upper Saddle River, NJ.

Rothaermel, F. T. 2001. "Incumbent's advantage through exploiting complementary assets via interfirm cooperation." *Strategic Management Journal* 22 (6-7), 687-699.

Salkever, A. 2002. "Giant Steps for a Software Upstart." *Business Week Online,* May 16.

Samuelson, P., Davis, R., Kapor, M.D., Reichman, J.H. 1994. "A Manifesto concerning the Legal Protection of Computer Programs." *Columbia Law Review* 94 (8): 2308-2431.

Shafer, S. M., Smith, H.J., Linder, J.C., 2005. "The power of business models." *Business Horizons* 48 (3), 199 - 207.

Shapiro, C., Varian, H.R. 1999. *Information rules: a strategic guide to the network economy*. Harvard Business School Press, Boston.

Stabell, C.B., Fjeldstad, Ø.D. 1998. "Configuring value for competitive advantage: on chains, shops, and networks." *Strategic Management Journal* 19 (5), 413-437.

Teece, D. J. 1986. "Profiting from technological innovation: implications for integration, collaboration, licensing and public policy." *Research Policy* 15 (6), 285-305.

Teece, D. J. 1998. "Capturing value from knowledge assets: The new economy, markets for know-how, and intangible assets." *California Management Review* 40 (3), 55-79.

Tripsas, M. 1997. "Unraveling the Process of Creative Destruction: Complementary Assets and Incumbent Survival in the Typesetter Industry." *Strategic Management Journal* 18 (Summer), 119-142.

Välimäki, M. 2003. "Dual Licensing in Open Source Software Industry." *Systemes d'Information et Management*, January.

Välimäki, M. 2005. *The Rise of Open Source Licensing. A Challenge to the Use of Intellectual Property in the Software Industry,* Turre Publishing, Helsinki.

von Hippel, Eric. 2005. *Democratizing Innovation.* MIT Press, Cambridge, Mass.

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

West, J. 2006. "Does Appropriability Enable or Retard Open Innovation?" in H. Chesbrough, W. Vanhaverbeke, and J. West, eds., *Open Innovation: Researching a New Paradigm.* Oxford: Oxford University Press, pp. 109-133.

West, J., Dedrick, J. 2000 "Innovation and Control in Standards Architectures: The Rise and Fall of Japan's PC-98." *Information Systems Research* 11 (2), 197-216.

West, J., O'Mahony , S., 2005. "Contrasting Community Building in Sponsored and Community Founded Open Source Projects." *Proceedings of the 38th Annual Hawaii International Conference on System Sciences,* 196c.

Yin, R. K. 1994. *Case Study Research, Design and Methods,* 2nd ed. Sage Publications, Newbury Park, Calif.

Dimension	Primary Focus	Teece 1986	Chesbrough & Rosenbloom 2002	Shafer et al. 2005	Morris et al. 2005
value creation for a specific buyer	customers	innovation	value proposition	create value	1. How do we create value?
		n/m	market segment	*strategic choices*	2. Who do we create value for?
sustainable value capture through combination of internal and external factors	competitors and costs	*complementary assets*	*competitive strategy*	*strategic choices*	3. What is our source of competence?
		n/m	cost structure and profit potential	capture value; *also part of strategic choices*	4. How do we competitively position ourselves?
		appropriability regime			5. How do we make money?
value network to jointly create and capture value from the whole product	suppliers and complementors	contractual modes	value chain value network	value network	n/m
		contracted complementary assets	complementary assets		
strategy derived from business model	implementation	strategic decisions	competitive strategy	strategic choices	n/m

n/m: not mentioned

Italics indicate this cell reflects a secondary use of the authors' concept

Table 1: Summary of prior research on business models and their resulting strategy

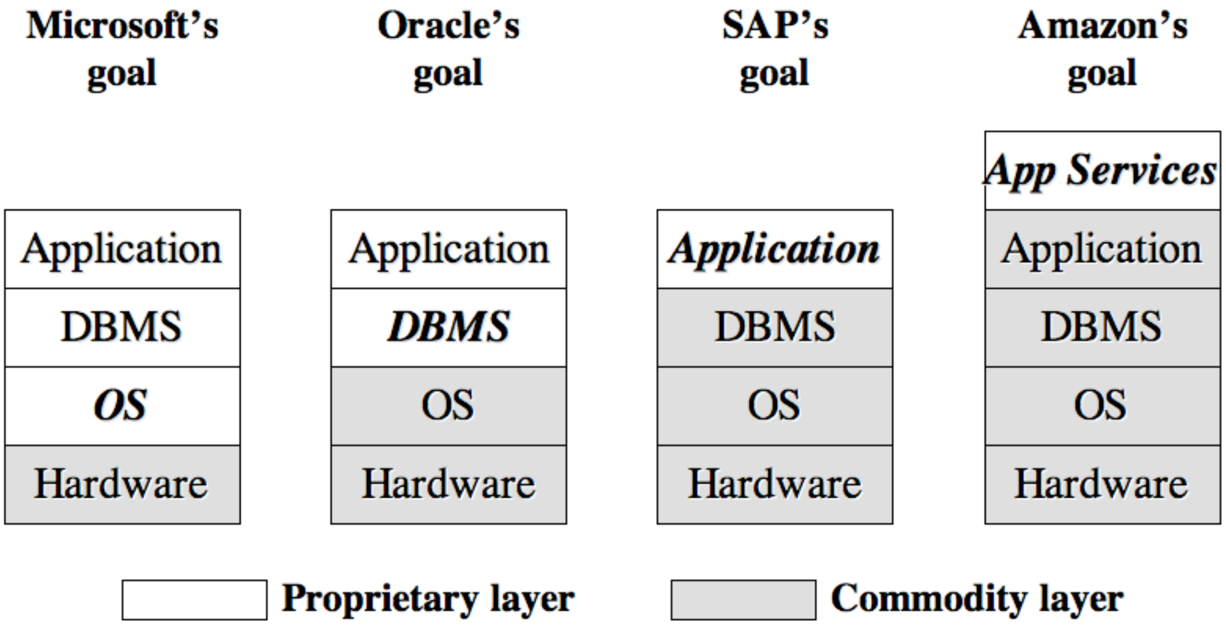

Figure 1: Commoditizing "up the stack"