Open Source and Standards

Steven Weber and David Lancashire
Berkeley Roundtable on the International Economy (BRIE)

INTRODUCTION

Information technology standards have historically been either fully proprietary or fully free. Open source standards pose an interesting challenge however, because they fall into a gap between the two: they restrict the freedom of their users even while providing them with extremely low-cost and reliable technology.

This raises interesting questions about how we should expect markets to standardize around open source products. If the nature of market competition in proprietary standards and free standards is modeled and predictable, does the same hold in markets faced with growth in open source standards? Is it possible to model and possibly predict where open standards will succeed and which firms will adopt them? This memo suggests that it may be so, and offers three hypotheses to guide a future research agenda:

1) Open Source Standards emerge at cross-industry chokepoints
2) Firm support for them is predictable and strategic
3) Differences between Development Communities matter

To build towards these hypotheses, the memo begins with a brief description of the four types of standards that exist in contemporary software markets: controlled, open interface, open and open source. A quick review of the basic economics of standardization provides a starting point for extending the logic of standardization into the open source space. Finally, we discuss the three hypotheses through the lens of the experiences of some large firms in the United States: Microsoft, Sun Microsystems and IBM.

SOFTWARE STANDARDS

We can differentiate four particular ownership patterns of standards in the software industry: controlled, open interface, open and open source standards.

Controlled standards are proprietary schemes for data storage and manipulation. They are controlled in the sense that use of the particular scheme is contingent upon the permission of the standard owner. Examples include Adobe's *Portable Document Format* and Autodesk's drawing system *Autocad* along with the vast majority of other dominant commercial software packages. In the case of Adobe, although the company makes software to read .pdf files available to users at no cost, those wishing to create such files or integrate Adobe software with third-party applications are required to pay substantial license fees. *Autocad* remains the dominant computer-aided-drafting tool in

the market, an impressive feat considering that commercial copies of the software sell for over $3000. In both of these cases a single firm controls the development of the software standard.

Open interface (or open API) standards are closely related to controlled standards, in the sense that they are proprietary schemes typically controlled by a single firm. What distinguishes them is that Open API standards are designed for interoperability and seek to encourage third-parties to link to their API or produce software dependent on their own. Cross-application interaction is encouraged by the availability of accurate programming interfaces and/or file formats. Typical examples in this category include the Microsoft Word .doc file format, as well as Sun Microsystem's programming language Java. Microsoft uses the open api to encourage developers to implement .doc in their own word processors, while Sun encourages developers to embrace its language by making the heavily-documented Java API freely available on the company website, and basic development tools available at no cost.

One of the consequences of this (moderate) shift towards integrating a broader developer community is that the process of standardization is somewhat more inclusive than that of proprietary standards. Because firms develop open APIs in order to encourage linking, the controlling firm has a vested interest in consulting with developers. Radical shifts are discouraged by the presence of a large community who exist in a symbiotic relationship with the standard-owner.

In contrast, *open standards* have no restrictions on use. No vendor holds veto power over the development of an open standard and anyone can develop applications that use it. One example of such a standard is the *Structured Query Language* (SQL) which migrated to ANSI in the late 1970s. The various *Requests for Comments* (RFC) published by the *Internet Engineering Task Force* (IETF) also qualify as open standards, as do web-standards such as HTML, TCP/IP, and such file formats as comma-separated values, postscript, rich-text format. Although there appears to be no definitive pattern to the development of these standards, the most successful have historically formed around inter-industry organizations and vendor-neutral groups like the *World Wide Web Consortium* (W3C).

Open Source standards are closest in spirit to open standards. In the narrowest sense, the term "open source" simply denotes software and/or data schemas for which the underlying source code is made available. In practice the term has come to mean software packages for which source code is *freely* available and governed by one of a number of *open source licenses*. The most prominent of these licenses is the GNU General Public License (GPL), and it is the one on which this memo restricts its focus.

Under the GPL, users are encouraged to modify and improve upon the original code base, but are obliged to release their improvements under the terms of the exact same license. In this sense, the GPL comes with strong restrictions on user freedom. It prohibits developers from incorporating GPL code in commercial projects, arguably reducing the commercial incentives to contribute to GPL products.

The development process of open source standards differs from other standards. Although there continues to be a strong debate over how open source standards are actually developed, the balance of evidence suggests that most development occurs through decentralized networks. Development is non-commercial in the sense that developers are rarely paid directly for their time and efforts. Instead, developers appear

attracted in the first instance to open source projects for a variety of personal reasons: personal connections with other developers, an abiding personal interest in the project itself, and/or feelings of solidarity with subsets of the open source community, as well as pressing individual needs for a solution to a problem that is not being offered by commercial alternatives.

STANDARDIZATION AND INFORMATION TECHNOLOGY

When do markets standardize around open source rather than other standards?

This is a difficult question to answer in part because it is unclear what standardization means in this context. Does it refer to cases where an open source product has dominant market share, or more narrowly only to those instances of industry-wide adoption?

While it is hard to imagine open source software ever fully conquering the market, it is moving in many cases to a *de facto* position of dominance in significant IT markets. The open source Apache web-server currently runs on 53 percent of all web-servers according to the most recent statistics from Netcraft. Apple's decision to integrate the Berkeley Standard Distribution (BSD) Unix into its OS X operating system has in some estimates made the open source-based BSD operating system three times more popular than desktop Linux in the United States, while in developing countries like China, Linux alone is estimated to have a 10 percent and growing market share. The International Data Group estimate that the free operating system will be run on 33 percent of all personal computers in Latin America by 2003.

The core economics of standards are simple. In information technology markets, standardization is encouraged by the heavy investments firms and individuals must continually make in non-fungible human and physical capital specific to very particular methods of data storage and manipulation. Cross-market connections also encourage standardization. Software firms must target their products to owners of specific operating systems. Manufacturers of those operating systems design them for particular combinations of processors and chipsets. Producers of the underlying hardware rely on common electronic interfaces to ensure market demand.

Reduced to its fundamental economic mechanisms, standardization can be viewed simply as a coordinating mechanism that encourages firms to streamline production in ways that create mutual gain. The existing economic literature categorizes these gains as resulting from *network effects*, *market-mediated effects* or *competitive effects*.

Network effects refer to situations where the value of any good scales positively with the number of people using it. Typical examples include automated teller machines (ATMs), telecommunications devices like telephones and fax machines, the QWERTY keyboard and most modular electronics hardware. Network effects are distinct from but often confused with *economies of scale*. The latter refer to situations in which the marginal *costs* of production fall as production size increases. The former focus on those benefits created by widespread adoption of a product or standard.

Market-mediated effects refer to benefits created by indirect forces such as product complementarity. Increases in the sales of hardware devices such as CD-Burners increase the demand for software capable of using the added features of these devices, for

instance, as the two products are complementary. These types of positive relationships between the markets for related goods exist all across the supply chain. By drawing an increasing number of software firms into providing packages for any one operating system or hardware features, hardware sales indirectly increase the very attractiveness of their product. Another example of a *market-mediated effect* might be the indirect benefit to users of thicker markets in the sale of used components.

Competitive effects are in many ways the most obvious. By encouraging producers to harmonize production around largely interchangeable products, standardization stimulates price competition between producers and lowers the end cost of ownership to consumers. A typical example here might be the current over-saturation in the market for Dynamic Random Access Memory (DRAM).

These three effects are not exclusive. In their attempt to understand the dynamic of the American computing industry during the 1980s, for instance, Jeffrey Church and Neil Gandal describe it as exhibiting "complementary network externalities". They see steady innovation across the industry as having transformed complementary goods into ones exhibiting increased network effects. Accelerating innovation in networking and data communications transformed the personal computer from a device primarily valued for spreadsheet calculations and word processing to one primarily valued for connectivity. A steady fall in the in the real cost of processing power enabled a host of new software applications. All of the above forces increased the value of the personal computer phenomenally while driving down its end cost.

It is sometimes difficult to distinguish between these three types of standards effects. *We believe that making the distinction is vital however, as each type of benefit encourages firms to engage in different forms of rent-capture, different types of standard-adoption, and to form different types of alliances for or against open source standards. Our research agenda is here to identify and explain these patterns.*

As Shapiro and Katz (1983) point out in their seminal article defining network effects, the presence of network externalities in any product may allow firms to capture monopoly rents over the long-term by inducing others to adopt their products as standards. The result is a sub-optimal allocation of social resources: consumers get "locked-in" as the market cannot support competing standards. Shapiro and Katz further demonstrate that in markets lacking perfect information *or* characterized by strategic actor behavior, the presence of an existing standard benefiting from network effects may also prevent the market from migrating to a more efficient standard. For example, the presence of low cost modem technology coupled with a highly-competitive ISP base in the United States arguably has hindered the rollout of broadband in America in ways it has not in Korea or the United Kingdom. Paul David's controversial paper about the relative efficiency of the QWERTY versus DVORAK keyboards contains a similar argument. It is useful to note in passing that sensitivity to this aspect of standardization may encourage certain types of public policy and thereby reinforce certain types of standardization. As Hemenway (1975) reports, recognition of this possibility inhibited the National Bureau of Standards from imposing early interface standards on the American personal computer market.

In industries with strong network effects we therefore expect firms to attempt to capture the rents created by long-term "lock-in". Firms should engage in highly competitive struggles to establish their own products as standards, and shift to a defensive

strategy once their product has achieved "critical mass" in the market: the threshold at which network effects accelerate.

Market-mediated effects also encourage strategic behavior, by allowing the producers of goods in less competitive industries to capture some of the value created in more competitive complementary industries. In their analysis of the pricing decisions of hardware firms in the 1980s, for instance, Chou and Shy (1990) demonstrate that the increased availability of third-party software led hardware manufacturers to increase their prices. This benefit directly benefited hardware producers while only exacerbating price competition among software producers.

In addition to influencing the distribution of rents in markets, differences in the nature of standardization create different expectations about when we expect standards to develop. Church and Gandal (1990) argue, for instance, that standardization occurs naturally in markets where the *network effects* of standardization are larger than the *competitive effects* of doing so. It is interesting to note (and we will return to this point) that this argument assumes that standardization is driven by consumer purchasing decisions. Other plausible hypotheses are that *market-mediated effects* encourage producers to form collaborative development relationships (through such organizations as the W3C), while *competitive effects* nurture price competition and create incentives for producers to fork standards by adding value-added features or proprietary extensions.

EXTENDING HYPOTHESES ABOUT OPEN SOURCE STANDARDS

Our initial survey of the American IT industry suggests the following hypotheses for where open source standards thrive, and how firms react to their appearance in the industry.

1. Open Source Standards Thrive at the Chokepoints of Complementary Products:

The decentralized nature of open source development restricts successful development to projects with a large potential user base. Although commercial support for open source projects mitigates this need to a certain extent, at present open source projects successfully competing with commercial projects seem restricted to those popular enough to attract a large group of developers and/or supporters.

These standards also develop overwhelmingly at mid-stream chokepoints where commercial software either is perceived as overpriced and unstable for the functionality desired by developers in its complementary industry, or is simply nonexistent. The user-facing and freely-available portable document reader *Adobe Acrobat*, for instance, has successfully avoided open source competition to date despite its extremely broad user base. We suggest that this is because the mere visual presentation of a file is a service with a high degree of complementarity. Where open source competition exists in this space is in the electronic manipulation and generation of .pdf files. Other projects are open source PDF-to-HTML converters.

The most successful open source projects that become industry standards are similar tools that manipulate data rather than simply present it. They create commercial opportunities instead of undermining them. Successful projects here include Linux (an

operating system), the GIMP and GTK (a graphics manipulation package and window toolkit), OpenOffice (an office suite) and computer languages such as Python and Perl.

Where open source development lags is in extremely specialized markets such as those for embedded operating systems, high fidelity audio processing, or sophisticated mathematical and statistical analysis packages. The most significant cause of slow development here is probably the lack of a broad user base. The highly-specialized expertise necessary to produce useful software in these areas also reduces the potential size of any developer community. It is also noteworthy that extremely specialized software of this type tends by virtue of its niche position in commercial markets not to occupy a major chokehold position between industries.

2. Firm Support for Open Source Standards is Strategic and Predictable

The support that firms give to open source standards depends on their competitive position in the marketplace. In any given sector, we expect that the dominant firms will prefer controlled standards for the reasons given by Shapiro and Katz. Conversely, niche actors should gravitate towards open API standards. In cases where dominant players are benefiting from existing network effects, new competitors will be unable to compete on costs alone. These firms will attempt to be competitive by encouraging new types of *network effects* and *market-mediated effects* to increase the value of their products to consumers in less direct ways.

Support for open source development is expected to come from suppliers of goods in complementary industries. These actors are engaged in cross-industry struggles to capture rents created in complementary industries, and should be expected to support the creation of rent-destroying standards in related industries.

A brief review of the support of the "open source" initiatives of Microsoft, Sun Microsystems and IBM provides initial support for these hypotheses:

Microsoft:

As the dominant software producer in the packaged software market, Microsoft has steadily resisted the encroachment of Open Source software and standards. This is hardly surprising given the importance of the company's closed source Windows operating system to its revenue stream.

As Open Source standards including Linux have gained strength, Microsoft has responded by shifted slightly towards the Open API model, advocating a "shared source" approach in which the company would make its source code available for private developers. The company has resisted pressure to build Windows in a modular fashion – a step that would encourage open source production in components. It has also strategically expanded (sometimes at a loss) into new markets that threaten to undermine the competitive dominance of the Personal Computer in the home computing market. Early initiatives of these sorts include the company's attack on Netscape with its free release of Internet Explorer. More recently Microsoft has aggressively pushed into the

market for portable computing devices (WinCE), home gaming consoles (Xbox) and small-scale web development (Personal Web Server/IIS).

Microsoft *does* support the development of open source standards in areas that threaten to undermine its competitors. Its support for the open source "Kaffe" implementation of the Java virtual machine is one significant case.

Sun Microsystems:

Sun Microsystem's most prominent foray into the open standards market has been through its support of the Java programming language with its promise of "write once, run anywhere". Designed as an alternative language specifically tailored for the easy creation of graphical user interfaces (GUI), the language aims to offer developers the opportunity to produce code that will run reliably in a range of computing environments. Thus, Sun has made just-in-time compilers and virtual machines for the execution of Java code freely available for most of the popular platforms (Microsoft Windows, Solaris, Linux, etc), and sought early-on to integrate its technology into the most popular web-browsers.

As a niche competitor whose profitability is still tied to software provision, Sun promoted Java as an Open Interface (or Open API) standard. The full specifications of the language are documented and published on the Sun website, while the company guarantees reliable cross-platform execution of identical code. These steps are obviously designed to make the language more attractive to developers while simultaneously increasing the value of the language by spurring the development of a large number of third-party applications written in the language.

Sun has also experimented with Open Source in areas designed specifically to weaken the market power of its rival and otherwise increase the attractiveness of Sun offerings. It has released a toolkit designed to teach Java programming (Jace) under an open source license, and is a prominent backer of the OpenOffice initiative: a project to create a full featured, cross-platform and MS Office-compatible suite of productivity tools.

IBM:

IBM is one of the most aggressive champions of "open source standards" in the American IT industry. To date, the company has spent more than $40 billion directly and indirectly in funding the development of open source software, the majority through the release of privately-developed and java-based software development tools to the public under an open source license.

This support is driven by the company's strategic interest in wresting a larger chunk of the web services market from Microsoft. IBM supports development of open source projects such as Linux, Apache, Jikes, and the Journaled File System as these goods are *complements* not *substitutes* to the company's major product offerings. The availability of reliable and low-cost software on non-Microsoft platforms encourages developers to use proprietary IBM offerings. The company's release of tools under its Eclipse program is intended primarily to make it easier for providers of web content and services to develop their goods on non-Microsoft platforms.

3. Developer Communities Matter:

While most firms attempt to leverage *network effects* in support of their preferred standard, the target communities of their strategies to do so often vary considerably. This makes it difficult to develop a unified "rational-actor" style explanation for standards competition. It also explains why simple economic models tend to fail in explaining the apparent selectivity of open source adoption. Communities made up of large software firms behave differently than those composed of smaller application developers. Consumers have different incentives than software providers.

Because different communities have diverging interests, it is necessary to understand the distinctions between them in order to make predictions about the potential success of open source standards. It is possible that different communities may be more and less effective in promoting the adoption of certain types of standards. Consumers may play a critical role in determining the dominant software packages in retail markets, for instance, but only a trivial role in influencing the underlying trajectory of innovation in hardware markets. Different types of commercial strategies may succeed or fail as result of targeting specific communities.

Microsoft's current strategy is based on a view of network effects in the IT industry as created by major software providers and key applications. The firm believes with some historic justification (in particular from its battle with Apple over the desktop market in the 1980s) that it can maintain dominance in the desktop space by forming close relationships with the producers of key consumer products. It is noteworthy that the company's "shared source" initiative, an effort to encourage developers to use Microsoft code, engages the producers of major application packages for the Windows platform, *not* smaller firms, individual developers or end-users of Windows. (Typical partners have to date included such large corporations as Adobe and engineering research groups at numerous universities.) The company expects that its control over the desktop space will allow it to extend its influence into tangential markets such as embedded operating systems, web-services, consumer gaming markets and others.

In contrast, Sun Microsystems is seeking to foster *network effects* from a community of smaller, independent developers. Its encourages these developers to contribute to the Java standard through the production of new programs, libraries, sample code and even helper applications, *but not through the alteration or reverse-engineering of the Java interpreter and/or compiler itself*. In sharp contrast to Microsoft, the company views low-level developers as the driving force behind technological change in the IT industry. It believes that innovation in the web-services market is driven by the producers of new services, not by the consumers of them.

Like Sun, IBM views technologies and standards in the IT industry as driven more by developers than consumers. The company's support for projects like Linux and its release of the Eclipse tools under an open source license are attempts to ensure that developers have an attractive and cost-efficient alternative to relying on Windows-based integrated development environments (IDE). Because the company is more active in complementary industries such as software services than Sun or Microsoft, its involvement in open source has taken the form of actual source code provision. By

releasing the underlying code of its Eclipse workshop, IBM hopes to encourage individual developers to integrating new features and functionality into the basic IDE. The developer community targeted here is more decentralized than that wooed by Sun.

Which standard ultimately "wins out" is likely to depend heavily on which communities are most important in any particular industry, and whether firms manage to successfully mobilize them in support of preferred standards. Microsoft believes that large software providers drive standard adoption. Sun and IBM are betting that standardization will be driven by communities of smaller software developers. Each strategy attempts to mobilize different types of supporting communities, and create different types of network effects.

HOW OPEN SOURCE STANDARDS ALTER MARKETS

The next stage of this research needs to develop hypotheses about how markets will change as open source standards emerge in some existing markets.

We expect dominant producers to resist the entry of these standards by leveraging their existing market power to reduce the importance of open source technologies. Strategies to achieve this include the zero-cost provision of certain services, or the extension of controlled-but-liberalized source code measures such as the Microsoft "shared source" initiative. Niche market players will promote open source development to the extent it does not undermine their ability to earn rents in their particular markets. Producers of goods in complementary industries can be expected to champion open source products.

Differences between developer communities will continue to matter. Microsoft will likely continue to focus on partnerships with large applications producers rather than smaller developers. Those firms such as IBM and Sun interested in targeting commercial application developers will conversely seek to create incentives geared to generate support from those actors. As underlying technologies change, the market – defining power of these communities will shift. That may become a significant determinant of which standardization strategies succeed.

The success of any initiative will also depend heavily on the position of that initiative in the market: and the extent to which it occupies a critical chokepoint between two industries, has a large potential user base, and is interesting enough to attract developers.

Reflecting on these three forces, the most important and most uncertain driving force in the current software market appears to be the likely behavior of second-tier developers (many of whom are currently wedded to open-API standards.) According to our hypotheses, we expect these players to support open source standards when these standards complement rather than compete with their own products. At present, with such extreme uncertainty over the profitability of open source-based business models, it is difficult to predict one way or the other how these smaller firms will behave.

SELECTED BIBLIOGRAPHY

Arthur, W.B. "On Competing Technologies and Historical Small Events: The Dynamics of Choice under Increasing Returns" Mimeo, Stanford University, 1983.

David, P.A. "Understanding the Economics of QWERTY, or Is History Necessary?" Mimeo, Stanford University, 1984.

David, P.A. "Clio and the Economics of QWERTY" *The American Economic Review*, Vol 75, No. 2. *Papers and Proceedings of the Ninety-Seventh Annual Meeting of the American Economic Association* (May, 1985), pp. 332-337.

Liebowitz, S.J. and Stephen E. Margolis "The Fable of the Keys" *Journal of Law and Economics*, Vol 33 (April 1990) http://wwwpub.utdallas.edu/~liebowit/keys1.html

Hemenway, D. Industrywide Voluntary Product Standards. Cambridge: Ballinger, 1975.

Kindleberger, C.P. "Standards as Public, Collective and Private Goods." *Kyklos*, Vol 36 (1983), pp. 377-397.

Farrell, Joseph and Garth Saloner, "Standardization, Compatibility, and Innovation" *The Rand Journal of Economics*. Vol. 16 (1985), pp. 70-83.

Church, Jeffrey and Neil Gandal. "Network Effects, Software Provision, and Standardization" *Journal of Industrial Economics*, Vol 40 (1992), pp. 85-103.

Chou, C.F. and Shy O. "Do Consumers Always Gain When More People Buy the Same Brand", SUNY-Albany mimeo. (1990)

"Sun Community Source Licensing" Sun Microsystems. <http://wwws.sun.com/software/communitysource/>

Gabriel, Richard P. and William N. Joy. "Sun Community Source License Principles" Sun Microsystems. <http://www.sun.com/981208/scsl/principles.html>

Hills, James "Linux spreading rapidly in Latin America" Linuxworld. April 23, 2001. <http://www.itworld.com/Comp/2362/LWD010424latinlinux/>

Romer, Paul. *"Are Nonconvexities Important for Understanding Growth?"* The American Economic Review, Vol. 80, No. 2, Papers and Proceedings of the Hundred and Second Annual Meeting of the American Economic Association. (May, 1990), pp. 97-103.

Romer, Paul M. *"Capital, Labor, and Productivity"* Brookings Papers on Economic Activity. Microeconomics, Vol. 1990. (1990), pp. 337-367.

Orlowski, Andrew. "BSD '3 times as popular as desktop Linux'" The Register. February 12, 2002. <http://www.theregister.co.uk/content/4/24060.html>

Sellers, Daniel. "Apple Could Double Market Share" MacCentral. May 24, 2002. <http://maccentral.macworld.com/news/0205/24.jobs.php>

Kim, Nam Hyung. "DRAM market may see stability toward end of 2002" Global Sources. January 16, 2002. <http://www.globalsources.com/MAGAZINE/EC/VUPOINT/VIEW02B.HTM>

Sandeep Junnarkar, "IBM makes $40 million open-source offer" CNET News. November 5, 2001. <http://news.com.com/2100-1001-275388.html>

Weber, Steven. The Success of Open Source. Forthcoming from Harvard University Press.