Paper Prepared for Presentation at
AALS Mid-Year Workshop on Intellectual Property
Vancouver BC
June 14-16, 2006

Title:
Open Source and Competition in the Software Industry

by
Pamela Samuelson,[*]
Richard M. Sherman Distinguished Professor of Law,
Boalt Hall School of Law, University of California at Berkeley

A useful lens through which to view open source and competition is to consider the competitive strategy that led IBM to adopt Linux as a platform and to donate valuable proprietary software to Linux under terms of the Free Software Foundation's General Public License (GPL).

Twenty years ago, IBM Corp. was the most vigorous advocate of (very) "strong" intellectual property rights for computer programs. Without strong copyright protection for programs, IBM contended, there would be insufficient incentives for firms to invest in software development. Its senior executives and lawyers contended that copyright law protected the structure, sequence, and organization of computer programs, as well as program code; that interface specifications were among the original elements of computer programs that copyright did and should protect; and that reverse engineering of computer programs for purposes such as achieving interoperability constituted copyright infringement.[1] IBM also relied on patents, trade secrets, licensing and technical measures to protect programs from unauthorized uses. IBM did not at that time publish source code, but rather distributed programs in machine-executable form and regarded all program internals, including interfaces, to be highly valuable trade secrets.

Virtually no intellectual property professional in 1986 would have predicted that two decades later IBM would embrace "free" or open source software and Richard Stallman's "copyleft" GPL. Yet, it has happened. IBM now spends $100 million a year on Linux and other open source software development projects.[2] The estimated value of the AIX software IBM contributed to Linux to strengthen the latter's ability to support enterprise level services is $500 million. IBM also released on an open source basis a suite of Eclipse software tools to aid programmers in writing better code. IBM does not, however, open-source all of its software development. Indeed, IBM still makes about a quarter of its overall revenues and a somewhat higher proportion of its profits from

[*] I wish to thank Robert J. Glushko and Mitchell D. Kapor for their insightful comments on this paper and Nate Anthony for research assistance.

[1] See, e.g., ANTHONY L. CLAPES, SOFTWARE, COPYRIGHT AND COMPETITION (1989) and SOFTWARS: THE LEGAL BATTLE FOR CONTROL OF THE GLOBAL SOFTWARE INDUSTRY (1993).

[2] See, e.g., IBM Corp., *Open Source and Linux*, available at http://www-1.ibm.com/industry/opensource.shtm.

developing and licensing proprietary software. Still, IBM's decisions to contribute previously proprietary software to Linux under the GPL, to invest in further development of Linux and other open source projects, and to make a significant part of its patent portfolio available to support Linux developments are but a few of many indicators that the intellectual property landscape and the nature of competition in the software industry has changed dramatically in the past twenty years.

There are at least three stories one can tell about this shift. IBM's adoption of open source can be viewed: (1) as an anti-Microsoft strategy; (2) as a consequence of changed business models in the software industry; and (3) as a manifestation of an open innovation strategy for promoting faster and more robust technical advances. All three stories have some explanatory power.

The anti-Microsoft story might open by observing that IBM misjudged the competitive significance of operating system programs. Prior to 1978, IBM routinely distributed source as well as object code versions of programs and published extensive information about the internal structure of its programs so that customers and independent software developers could build upon the IBM platform, thereby creating further demand for the IBM platform. In the early 1980's, when IBM was the dominant firm in the industry, it began to restrict access to source code and interface information in order to bring compatible systems under its control. Contrary to its usual practice, IBM did not build a proprietary operating system when it decided to enter the newly emerging market for personal computers (PCs). Instead, its PCs contained operating system programs licensed from Microsoft. To enable and encourage a plentiful supply of programs for the PC platform, IBM required Microsoft to make interface information available to application developers. The IBM PC was a substantial success in the market and became an industry standard. But IBM's lack of control over the PC operating system enabled Dell, Compaq, and other vendors to offer equivalent technologies running Microsoft's operating system, thereby commoditizing the PC platform. Microsoft's shrewd business practices in licensing its operating system to PC developers enabled it to obtain monopoly power for its platform.

In the early to mid-1980's, IBM worked with Microsoft to develop a next generation set of operating system programs known as OS/2, although Microsoft was concurrently hedging its bets by developing Windows 3.x.[3] IBM launched a new line of personal computers under the PS/2 brand with OS/2 under the hood in 1987. After Microsoft's launch of Windows 3.0 met with considerable success in the marketplace, Microsoft decided to shift its primary applications interface away from the OS/2 model and worked towards launching Windows NT for workstations, minicomputers and other high end multi-user settings. This, among other things, strained relations between IBM and Microsoft and they ceased joint development of OS/2. IBM made further investments in OS/2, but neither PS/2 nor OS/2 was a success in the marketplace, in part because it was not compatible with programs written for Windows platforms. IBM's half a billion dollar bet on the highly proprietary OS/2 platform was not a complete loss, but

[3] For much of the relevant history, see CHARLES H. FERGUSON & CHARLES R. MORRIS, THE COMPUTER WARS: THE POST IBM WORLD (2002).

IBM did abandon OS/2 development more than a decade ago and is phasing out support for OS/2 customers. Microsoft's operating systems became and have remained de facto industry standards. [4]

Linux is the first operating system that holds real promise to challenge Microsoft's hegemony. By the time IBM adopted Linux as a platform, Linux already had a large installed base of customers and a large community of developers committed to contributing regularly to the ongoing development of Linux. Spending $100 million a year on Linux development is a bargain for IBM by comparison with developing a new operating system from scratch. Investing in Linux allows IBM to be independent from Microsoft, its licensing terms, and its willingness to reveal (or not) interface specifications for its platforms. Seemingly to counter IBM's investment in Linux, Microsoft helped to broker outside investments in SCO to aid its challenge of IBM's contribution of AIX source code to Linux as a breach of licensing agreements IBM had with AT&T as to UNIX, agreements to which SCO claims to be a successor in interest. IBM is vigorously defending the SCO lawsuit and is confident that it will succeed over time. In this story about open source, IBM's support of Linux is a move to defeat Microsoft's monopoly in the operating system market.

But IBM's adoption of Linux should not be understood as a pure anti-Microsoft move. IBM would not have committed itself to investing in open source unless it had developed business models to enable it to recoup investments in Linux and other open source projects. A second story about open source thus concentrates on evolving business models in the software industry.

Software was not at the core of IBM's initial business model. IBM initially developed software in order to sell hardware. Computers needed programs to be useful, and IBM provided programs to run on its computers to make the hardware more attractive to its customers. After IBM achieved a dominant position in the computer market in the 1970's and early 1980's, it sought to fend off competitors such as Fujitsu by controlling access to interface specifications. IBM also changed interfaces with some frequency, thereby frustrating efforts by competitors and developers of complementary products to achieve compatibility with IBM computer systems. Antitrust authorities in the US and EU challenged IBM's bundling of software and hardware and its modifications of interfaces that had the effect—and allegedly the purpose—of creating incompatibilities with non-IBM technologies as monopolistic practices.

From roughly the early 1980's through the mid-1990's, IBM's business model shifted somewhat away from hardware and toward proprietary software development and licensing (hence, among other things, IBM's decision to acquire Lotus Development Corp.). "Before 1969, IBM received virtually no revenue from software. By 1984, IBM's software revenue exceeded the total software revenues of all of its competitors combined. In 1985, IBM achieved a gross margin of 70% on software products, in

[4] The open source Apache platform has a dominant position in the webserver market, in part because of support from IBM Corp. See, e.g., STEVEN WEBER, THE SUCCESS OF OPEN SOURCE 6, 124-26 (2004).

contracts to a 55% margin on hardware."[5] IBM's shift toward software was partly due to recognition of the power of network effects in many software markets, especially for those deemed "killer apps." The more widely adopted mass-market software is, the more attractive it is for others to acquire the same software, particularly when the program is used to exchange information.

The "winner take all" nature of competition in many software markets gave IBM reason to believe that a well-designed software product (such as Lotus 1-2-3) could attain a monopoly position without undue difficulty and maintain that monopoly—allowing the firm to charge higher than competitive prices—for some time. Successful though this strategy was in some key market segments, it was undermined by a number of factors, including Microsoft's dominance in key platform markets, Microsoft's decision to bundle application programs into attractively priced packages (e.g., Microsoft Office), and Microsoft's incorporation of many previously separate program functions (e.g., browser functionality) into its operating system. In the mid- to late 1990's, the mass market for software became increasingly commoditized. The rise of free and open source software is partly a reaction to and partly a consequence of this commoditization.

IBM's shift toward open source is part of the legacy of Louis Gerstner, IBM's CEO in the 1990's, who thought IBM had the wrong attitude toward its customers and challenged the company to reconceive its business models. Gerstner reportedly observed to key IBM insiders: "This is the only industry where competitors don't regularly agree on standards to enable greater value for customers." To which IBM executives responded: "Let us explain about lock-in, network effects, de facto standards and the five ways to play." Gerstner's reaction was: "That's interesting … let me get this straight … you're telling me the strategy is to lock-in our customers and then gouge them on price."[6] Gerstner insisted that this was not what IBM should be about, and he set out to change IBM's business models and internal culture to create a more customer-centric business environment.

Finding out what customers want and providing products and services to support customer needs has become IBM's vision for its present and future. Customers, IBM discovered, want a sustainable and reliable software eco-system, open standards, interoperability, and customization tailored to their needs. IBM now conceives of Linux as part of the sustainable information eco-system that serves customer needs better than a proprietary operating system. IBM has two strategies for profiting from open source, and in particular, from Linux: "First, open source software is by some measures less expensive than proprietary software, so using it lowers the overall cost a customer pays for IBM's computers, applications, and services. Second, it provides a common platform on top of which IBM can build and sell special applications and services.[7]" IBM's business model now focuses on selling high end hardware, proprietary software running on top of Linux, and integration and other customized services to enterprise customers.

[5] JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL: INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY 25 (1995).

[6] Conversation with Dan McGrath, IBM Director of Corporate Strategy, Sept. 26, 2005.

[7] "An Open Secret," The Economist, Oct. 22, 2005, p. 13.

To support its new business models, IBM has recrafted its intellectual property policy and made key strategic moves such as dedicating portions of its patent portfolio to a patent commons in support of Linux.

IBM's are not the only business models that can be used to support open source development. Some charge fees for installation of open source software, customization for special purposes, maintenance of the software, warranties, and other services. Some firms, such as MySQL, offer a basic level of program functionality on an open source basis and more advanced functionality on a proprietary basis. Once customers have experience with the basic version, some may decide to switch to the enhanced version to get the extra features. Customers won't fear lock-in because the basic functionality is available on an open source basis, and they can, if they wish, develop a "fork" in the code by making changes to the open source version, creating their own enhanced functionality as an alternative to the proprietary version if it doesn't meet their needs or is offered on terms customers find unattractive.

Open innovation is the third story that can be told about the rise of open source. The more open the environment in which innovation takes place, the more cheaply, quickly, and widely it can be disseminated and used. The traditional proprietary model of software development takes place in a closed environment in which all costs of development are borne by the producer, costs that the firm will be unable to recoup if the producer loses out in a "winner take all" market. Even if a proprietary developer attains a monopoly position in a software market, success may not be as long-lived as the firm might hope, for competitors or providers of complementary products or services will often find ways to achieve compatibility and undermine the monopolist's control of the market, or a next generation product will leapfrog over the previous year's winner. To retain its customers, moreover, a proprietary developer must invest a lot of money in support services, and to keep customers buying, proprietary developers have to invest in ongoing evolution of the product, upgraded versions of which customers may or may not want to buy, and to maintain compatibility with prior versions.

Among the key advantages of open source is that the difficulties and costs of software development can be distributed among many contributors. IBM may be contributing $100 million a year to the development of Linux, but firms such as Nokia, Intel, and Hitachi are making substantial investments as well. Commercial investments in Linux are estimated to exceed $1 billion a year. Sizeable though its contribution is, IBM is sharing with others the effort and expense of developing this core infrastructure. Because Linux had a substantial customer base when IBM decided to adopt it, IBM avoided one substantial cost typically borne by proprietary software developers, namely, launching a vigorous marketing campaign to induce firms and individuals to purchase a new or updated product.

The open source development model attracts follow-on developers to build upon the open source base because interfaces are easily discerned, thus overcoming a common barrier to entry for independent service vendors who develop software for proprietary platforms. Open source also means that one's customers can become part of the development team, willing to invest time, money and energy on making the software better (e.g., fixing bugs), more robust, and more extensible. There is less need for internalizing support services; customers can either fix the code themselves, or pay for support as a value-added service to the open source product.

Improvements in software are, moreover, shareable when the software has been developed under the GPL. Regular updated releases of Linux can disseminate improved versions to the world via the open innovation environment known as the Internet. IBM and other firms can take advantage of ongoing open innovation done by others on Linux and other GPL projects. They can see what that innovation is because the GPL requires publication of source code of derivative programs of GPL software. By studying others' innovations, IBM engineers may perceive opportunities for further refinements to the code or for new technologies that can be built on the open source base. IBM has embraced open source in part because its open innovation model of development brings about more innovation and a faster pace of innovation. [8]

"The essence of open source," Steven Weber has suggested, "is not the software. It is the process by which software is created."[9] That is, open source software is the product of a different kind of production process than is used to make proprietary software. To enable software to be developed by a decentralized collaborative process, programs must be broken down into discrete modules so that different people can work on different modules at different times without loss of coherence to the whole. Because of the very different production process by which it is made, proprietary software cannot readily be open sourced. When Netscape, for example, released its browser code under an open source license after losing much of its market share to Microsoft, the code had to be substantially rewritten and restructured in order to make it amenable to open source development processes. Distributed collaborative development of software makes sense because of the growing complexity of information technology, the dispersion of talented engineers who use the Internet to collaborate and enjoy doing so, and the need to integrate software from many sources to enable, for example, global supply chains to operate efficiently. Modularization of code has made software and its component parts more interchangeable, and created opportunities for niche market players to reassemble components to make new products and services. IBM is promoting the emergence of a field of "services science" to study in a systematic way the assembly of services from service components.

Open source has had some important successes, but the proprietary side of the software industry is not going to wither away and die. Rather, it seems likely that proprietary and open source software will continue to coexist, as they do now. IBM is one of thousands of firms that mix open source and proprietary business models, and more experiments with mixed models are to be expected. Yet, it is worth recognizing that open source is having competitive effects on firms that retain fully proprietary models. Microsoft's Internet Explorer is still the dominant browser software, although

[8] See, e.g., Henry W. Chesbrough, *The Era of Open Innovation*, 44 MIT Sloan Mgmt Rev. 35 (Spring 2003).

[9] Weber, supra note 4, at 56.

the open source Firefox has somewhat eroded Microsoft's share of this market. After Firefox introduced innovations in its browser, Microsoft responded with innovation that it might not have undertaken in the absence of this competition. Now that Firefox has made a deal with Google to feature a Google search box in its browser, Firefox has an assured source of revenue it can use to fund further open source development. Competition from open source has also affected Microsoft's willingness to be somewhat more open with source code than it was in the past. Open source is thus having significant effects on the dynamics of competition in the software industry.

The success of open source does not signal a lessened importance for intellectual property rights, but rather a shift in the way these rights are employed in the software industry. Open source licenses condition a user's legal right to do certain things with copyrighted software, such as to adapting it, on the user's acceptance of the terms of the license. In this model, both copyright and licenses are used in a significantly different way than proprietary firms use these same forms of legal protection. Trade secrecy still plays an important role in many software markets, but open source publication of source code and disclosure of interfaces are far more common today than two decades ago. Formation of patent pools to support open source development (sometimes described as "patent commons") is yet another shift in the use of intellectual property rights in the software industry.

It remains to be seen how courts will adapt existing intellectual property doctrines to accommodate open source and open innovation concepts. Who, for example, is the "author" of large distributed collaborative open source projects such as wikipedia? Is it possible to make fair uses of open source code in a proprietary product? Will courts enforce efforts to revoke open source licenses if firms purchase rights in an open source product and want to convert a previously open source resource to which many people contributed into a proprietary product? Will customary norms of open source communities be given weight when disputes arise over, for example, an effort to fork the code and designate a successor program as the true embodiment of a trademarked name for an open source project? To what "derivative works" does the GPL attach? These are but a few of the many issues that intellectual property scholars will have the opportunity to explore in coming years as the open source phenomenon continues to change the competitive landscape of the software industry.