

Forthcoming 49 Communications of the ACM (Oct. 2006)  
Legally Speaking:  
IBM's Pragmatic Embrace of Open Source

by  
Pamela Samuelson

Twenty years ago, IBM Corp. was the most vigorous advocate of (very) “strong” intellectual property (IP) rights for computer programs. Without strong copyright protection for programs, IBM contended, there would be insufficient incentives for firms to invest in software development. IBM's executives and lawyers asserted that (1) copyright law protected program code from copying and redistribution, but also the structure, sequence, and organization of programs; (2) interface specifications were among the original elements of computer programs that copyright did and should protect; and (3) reverse engineering of computer programs for purposes such as discerning interface information in order to develop interoperable programs infringed copyrights.[4]

IBM also relied on patents, trade secrets, licensing and technical measures to protect programs from unauthorized uses and modifications. IBM did not at that time publish source code, but rather distributed programs in machine-executable form and regarded the internal details of program design, including interfaces, to be highly valuable trade secrets.

At the opposite end of the IP spectrum from IBM back then was Richard Stallman who in 1985 published the GNU Manifesto, which argued on moral principles that software source code should always be publicly available and that anyone should be able to adapt software and to share the original or adapted software with others. Stallman developed the General Public License (GPL) to instantiate his “copyleft” norms.

Who in 1985 would have guessed that two decades later IBM would embrace “free” or open source software and the GPL? Yet, it has happened. IBM now reportedly contributes \$100 million a year to the development of Linux and other open source software projects. IBM donated some components of its proprietary AIX software, the IBM flavor of Unix, to Linux to strengthen the latter's ability to provide enterprise-level capabilities and scalability. IBM also released on an open source basis the Eclipse software tools suite and framework and contributed resources to start an open source consortium to support and extend it.[3]

IBM does not, however, open-source all, or even most, of its software. Indeed, IBM still makes about a quarter of its overall revenues and a much higher proportion of its profits from developing and licensing proprietary software. Nevertheless, IBM's decisions to embrace Linux as a platform, to invest in further development of Linux and other open source projects, and to make some of its patent portfolio available to support Linux development are indicators that the intellectual property landscape and the nature of competition in the software industry have 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.

## ANTI-MICROSOFT STORY

Prior to 1978, IBM routinely distributed source as well as object code versions of programs and published extensive technical information about the internal structure of its programs so that customers and independent software vendors (ISVs) could build programs for IBM computers, thereby fueling demand for them.[1]

In the early 1980's, when IBM became the dominant firm in the industry, it began to restrict access to source code and interface information in an attempt to bring compatible systems under its control.

Contrary to its usual practice, IBM decided not to build a proprietary operating system when it entered the newly emerging market for personal computers (PCs). Instead, its PCs came loaded with operating system programs licensed from Microsoft, then a small privately-held firm. 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 wildly successful and became an industry standard.

But IBM's lack of control over the PC operating system helped to enable Dell, Compaq, and other vendors to offer equivalent technologies running Microsoft's operating system, all of which interoperated with software created for the IBM PC. The effect was to commoditize the PC platform. Microsoft's shrewd business practices in licensing its operating system to PC developers and to encourage network economies 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.[5] 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 phenomenal 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.

Neither PS/2 nor OS/2 was a success in the marketplace, in part because OS/2 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 IBM

abandoned OS/2 development more than a decade ago and is phasing out support for OS/2 customers. Microsoft's platforms became and have remained de facto industry standards.

Linux is the first operating system that holds real promise to challenge Microsoft's dominant position. By the time IBM adopted Linux, it already had a large installed base of customers and a large community of developers committed to contributing regularly to its ongoing development. 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's licensing terms and willingness to reveal (or not) interface specifications for its platforms. IBM's support of Linux increases the chances that Linux will succeed in its competition with Microsoft in OS markets.

## BUSINESS MODELS STORY

Software was not at the core of IBM's initial business model. IBM initially developed software mainly in order to sell hardware. Computers needed programs to be useful, so IBM provided some programs to run on its computers to make the hardware more attractive to its customers.

After IBM achieved a dominant position in the computer industry 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 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 contrast to a 55% margin on hardware." [1]

IBM's shift toward software was partly due to recognition of the power of network effects in many software markets, especially for "killer apps." The more widely adopted mass-market software is, the more attractive it may be 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 platform 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. Embracing Linux has also brought IBM significant public relations benefits.

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.

## OPEN INNOVATION

Proprietary software development takes place in a closed environment in which the 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 or if demand for the software is modest. 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 in support services. To keep customers buying, proprietary developers must engage in ongoing evolution of the product, upgraded versions of which customers may or may not want to buy, and maintain compatibility with prior versions.

Among the key advantages of open source is that the difficulties and costs of designing, developing, and improving software can be distributed among many contributors. IBM may be spending \$100 million a year on 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 some substantial costs typically borne by proprietary software developers, namely, launching a vigorous marketing campaign to induce firms and individuals to purchase a new or updated product.

Open source development makes it easy for firms to build upon the open source base because interfaces may more easily be discerned. This overcomes a common entry difficulty for ISVs 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 for open source products.

Improvements are, moreover, shareable when software has been developed and distributed under the GPL. Updates of Linux are regularly disseminated to the world via the open innovation environment known as the Internet. IBM can take advantage of ongoing open innovation done by others on Linux and other GPL projects because the GPL requires disclosure of source code of derivative programs of GPL software. By studying others' innovations, IBM engineers may perceive opportunities for building new technologies on the open source base. Some believe the open innovation model facilitates a faster pace of innovation.[2]

"The essence of open source," in Steven Weber's view, "is not the software. It is the process by which software is created." [8] Weber explains key differences between the production processes that yield open source and proprietary software. To enable the development of open source software in a decentralized collaborative manner, 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 makes sense because of the increased 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 more 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.

## CLOSING THOUGHTS

IBM’s substantial investment in open source is one of many indicators that the nature of competition in the software industry has changed significantly in the last twenty years. Although open source has had some important successes—and IBM has contributed to some of them—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 in the works.

Even among firms whose business models are based exclusively on proprietary software, open source is having noticeable competitive impacts. Microsoft now seems more willing to provide ISVs with access to source code than in the past. After Firefox introduced innovations in its browser, Microsoft made revisions to Internet Explorer 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, this open source project has an assured source of revenue it can use to fund further open source development.

The success of open source does not signal a lessened importance for intellectual property rights compared with twenty years ago, but rather a shift in the way these rights are employed in the software industry. Open source developers use copyright and licenses differently than proprietary firms use these same forms of legal protection. Trade secrecy still plays an important role in many software markets, but open source developers forego trade secret protection by publishing source code. Formation of patent pools to support open source development is yet another shift in the use of intellectual property rights in the software industry.

Twenty years ago, few would have predicted that open source would be a viable production process for designing and developing software and would yield products upon which many viable business models could be built. Intellectual property professionals are still somewhat puzzled by the rise of open source, but we admire creative uses of IP rights to advance the philosophy of open source and achieve success in the marketplace. While some software developers adopt open source as a matter of principle, IBM's embrace of open source seems more driven by pragmatism and its relative market power. Yet, IBM's embrace of open source is no less welcome for being pragmatically inspired.

Pamela Samuelson is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California at Berkeley.

Acknowledgements: Thanks to Jonathan Band, Robert J. Glushko, Mitchell D. Kapor and Steven Weber for their insightful comments on an earlier draft of this paper and Nate Anthony for research assistance.

Sources:

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

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

[3] Capeck, P.G., Frank, S.P., Gerdt, S., and Shields, D., *A History of IBM's Open-source Involvement and Strategy*, IBM Systems Journal, 44:2, 249 (2005).

[4] CLAPES, ANTHONY L. SOFTWARE, COPYRIGHT AND COMPETITION (1989).

[5] FERGUSON CHARLES H. & CHARLES R. MORRIS. THE COMPUTER WARS: THE POST IBM WORLD (2002)

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

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

[8] STEVEN WEBER, THE SUCCESS OF OPEN SOURCE (2004).