RED HAT'S COMMENTS TO THE JOINT FTC/DOJ HEARING ON COMPETITION AND INTELLECTUAL PROPERTY LAW

March 20, 2002

Red Hat, Inc.
Mark Webbink

with Colin Crossman, Thomas Griffin and David Silverstein

About the authors: Mr. Webbink is Sr. Vice President and General Counsel of Red Hat Inc. Messrs. Crossman, Griffin and Silverstein are law students at Duke University School of Law.

Red Hat would like to take this opportunity to give an open source perspective on the granting of software and business method patents, and the importance of preserving anti-trust causes of action in the patent context.

For starters, Red Hat takes the general position that patenting business methods and software applications is the wrong approach. We are of the view that patent protection for these two categories should be minimized and ultimately eliminated. With regard to the nexus of patent law and anti-trust, it is our position that competition law is necessary and beneficial to innovation particularly where large patent portfolio holders are deploying their patents to stifle competition in other markets.

One reason that patenting business methods and software applications is the wrong approach is because of the problems of updating, maintaining and accessing a database by which the patent office can search effectively for prior art. Unlike most other technologies - such as pharmaceuticals, chemicals, and industrial design - there are no extensive, comprehensive databases where software prior art can be reliably found. In the computing arts, particularly in the open source community, a great amount of innovation has been and is produced by individuals who never publish in industry journals. For almost 20 years commercial software development was protected as trade secrets, thus denying the general public of broad knowledge of that prior art. Even in the years since the USPTO has accepted applications for software and business method patents, the long periods between filing and publication have frequently exceeded the useful life of the process claimed. Then to, the publication of choice for open source innovation primarily occurs via email listservs or Internet chat rooms. While this free-form, decentralized approach to innovation is a hallmark of open source, it does not lend itself well to comprehensive and organized searches by the patent office.

Thus, diligent searches for business methods and software are often unreliable and costly.[1] This is particularly true given that a simple program can consist of hundreds of previously developed algorithms and business methods of unknown origin. As a result, software and business method patents are being obtained too easily and often contain well-known prior art, ideas that lack novelty to those skilled in the art, or concepts already in the public domain.[2]

These patents, a generous proportion of which would not survive judicial scrutiny, still receives a presumption of validity. Since financially strong companies can more easily acquire large patent portfolios which include such dubious patents, the burden typically falls to the public and small-scale innovators to consider expensive and time-consuming litigation. This problem is exacerbated by the fact that "often, open source developers don't realize their [own] ideas are patentable. And even if they do, they often don't have the time and money to patent them."[3] Unchallenged, financially strong companies holding large portfolios of software patents are able to extract unfair fees and leverage their patent monopolies in unintended ways.

The current regime of patent protection for business methods and software patents is problematic for two other reasons. First, the speed of innovation in these areas is so fast that the long periods of protection granted by patents is stifling subsequent innovation. Second, software in particular can be adequately protected under copyright law.

The patent system is designed to "promote the Progress of Science and the useful Arts" by giving "Authors and Inventors" the exclusive right to practice their inventions for twenty years.[4] In the traditional areas of Chemistry, Biology, Physics, Medicine, Mechanics, etc., the twenty-year patent term allows the inventors to recoup their investments. Pharmaceutical innovations cost about $500 million and require twelve to fifteen years on average to bring to market.[5] In software and business methods, however, innovation proceeds at a much faster pace. Even in large software packages, such as Linux or Microsoft Windows, new versions are released at a fraction of the development costs and time compared to pharmaceuticals.

While it is our position that patent protection for software and business methods should be phased out, some of these problems can be relieved by shortening the patent term for software and business-method patents to three years. Three years is more in line with the development time and cost that software and business methods face. As it is, copyright law offers long-term, yet softer protection for software developers and does less to hinder follow-on innovation. Other software developers have taken advantage of the mild scrutiny of software patent applications to leverage their dominance into other areas.

Red Hat is also concerned with efforts to limit the scope of anti-trust actions where patents are involved. While the Supreme Court in *Eastman Kodak* has broadly indicated that "power gained through some natural and legal advantage such as patent, copyright, or business acumen can give rise to liability if a 'seller exploits his dominant position in one market to expand his empire into the next [market]'", the Federal Circuit appears to have narrowed the criteria by which anti-trust claims may be considered.[6] Although the *Xerox* case was more about a unilateral refusal to license intellectual property, as compared to a case of "tying" in *Kodak*, we are nonetheless concerned about situations where large portfolio holders are actually undermining competition by refusing to deal with particular customers in a fair marketplace.[7]

As stated earlier, we are generally opposed to the patenting of software and business methods. Allowing these patents to "trump" competition law only makes matters worse. There is a real question as to whether the Federal Circuit is following the Supreme Court's lead, given the former's position which tends to severely weaken anti-trust liability. As a policy matter, it makes it even more difficult for others to stop anti-competitive behavior on the part of large patent portfolio holders. As a few examples of consolidations and potential abuses of software patents, consider the cross-licensing between Microsoft and KYE Systems Corp., SGI[8], and Inprise[9].

The debate has gone on for more than a hundred years concerning where the balance should be found between the doctrines of anti-trust and patent law. However, commentators have noted that the rise of antitrust legislation in the 1890's in this country was partly in response to the reluctance of manufactures to license their patents. Even more so today, anti-trust law still acts as a check on strategic behavior on the part of patent holders who might otherwise extend their monopolies well beyond the innovation the conceptual framework was designed to protect.[10] The realization here is that patents should not be used to weaken or destroy the creation or maintenance of parallel or derivative markets, and the avoidance of such harm is properly within the purview of anti-trust law.

[1] A U.S. software patent will generally cost approximately $10,000 to prepare and file and, in our increasingly global community, it is necessary to obtain patent protection in all major marketplaces as well. This means filing a patent application in each jurisdiction that may constitute a major market for the software program. Costs to translate, prepare, and file corresponding patents will average about $5,000 per jurisdiction. http://www.redherring.com/mag/issue19/property.html (March 1995 Issue, last visited March 19, 2002).

[2] IBM patent #4,742,450 covers “shared copy-on-write segments, even though shared segments and copy-on-write have been used since the 1960’s. See also: Patent 4,956,809, issued to the Mark Williams company on September 11, 1990, applied for in 1982, covered storing data on disk in a machine-independent order, as the DVI, TFM, GF, and PK file formats. Even though TeX was "prior art”, the patent was granted. Since there is a strong presumption in the courts of a patent's validity once it has been granted, there is a good chance that users or implementers of TeX could be successfully sued on the issue. http://www.cl.cam.ac.uk/texinfodoc/eplain_8.html (last visited 3/19/02).

[3] http://www.linuxworld.com/linuxworld/lw-1998-11/lw-11-thesource.html

[4] U.S. CONST. ART. I, SEC. 8, CL. 8; 35 U.S.C. §§ 1-376.

[5] PhRMA “Why do prescription medicines cost so much?” http://www.phrma.org/publications/publications/brochure/questions/whycostmuch.phtml (last visited March 19, 2002).

[6] *Eastman Kodak Co. v. Image Technical Services, Inc.*, 504 U.S. 451, 479 (1992) citing *Times-Picayune Pub. Co. v. U.S., 345 U.S. 594, 611* (1953).

[7] *CSU, L.L.C. v. Xerox Corp. (In re Independent Serv. Orgs. Antitrust Litig.)*, 203 F.3d 1322 (2000).

[8] SGI made $62.5 million selling "non-core intellectual property rights" to Microsoft; see: http://zgp.org/linux-elitists/20011023171025.D16614@navel.introspect.html (last visited March 19, 2002).

[9] Microsoft poured $125 million into the troubled development-tools vendor and licensed Inprise's technology patents for $100 million; see: http://www.idg.net/idgns/1999/06/08/MicrosoftCashCouldBailOutAiling.shtml. (last visited March 19, 2002).

[10] For an example of predatory behavior arising from patent pooling and cross-licensing, see: http://www.google.com/search?q=cache:OejF7l2HwZgC:www.virtualwallstreet.com/news/Volume_4_Issue_8_Aug_2000/04_8_103.shtml+Microsoft+and+patent+cross+licensing&hl=en&ie=ISO-8859-1 (last visited March 19, 2002).