2007-1130
(Serial No. 08/833,892)

In The
United States Court of Appeals
For The Federal Circuit

IN RE BERNARD L. BILSKI and RAND A. WARSAW

APPEAL FROM THE UNITED STATES PATENT AND TRADEMARK OFFICE, BOARD OF PATENT APPEALS AND INTERFERENCES.

BRIEF OF *AMICUS CURIAE* RED HAT, INC. IN SUPPORT OF APPELLEE

Robert H. Tiller
Richard E. Fontana
RED HAT, INC.
1801 Varsity Drive
Raleigh, North Carolina 27606
(919) 754-4847

Counsel for Amicus Curiae Red Hat, Inc.

Dated: April 7, 2008

THE LEX GROUP^DC ♦ 1750 K Street, N.W. ♦ Suite 475 ♦ Washington, D.C. 20006
(202) 955-0001 ♦ (800) 815-3791 ♦ Fax: (202) 955-0022 ♦ www.thelexgroupdc.com

CERTIFICATE OF INTEREST

Counsel for the amicus Red Hat, Inc. certifies the following:

1. The full name of every party or amicus represented by me is:

   Red Hat, Inc.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

   N/A.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the party or amicus curiae represented by me are:

   Red Hat, Inc. has no parent corporation. More than 10 percent of the common stock of Red Hat, Inc. is held by both T. Rowe Price Associates, Inc. (a subsidiary of the publicly held corporation T. Rowe Price Group, Inc.) and Legg Mason, Inc.

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court are:

   Robert H. Tiller
   Vice President and Assistant General Counsel, IP, Red Hat, Inc.

   Richard E. Fontana
   Open Source Licensing and Patent Counsel, Red Hat, Inc.

April 7, 2008

Richard E. Fontana
Red Hat, Inc.

INTEREST OF AMICUS

Red Hat, Inc. is the world's leading provider of open source software and related services to enterprise customers. Its software products are used by Wall Street investment firms, hundreds of Fortune 500 companies, and the United States government. Headquartered in Raleigh, North Carolina, Red Hat has offices in 58 countries.

Red Hat's interest in this proceeding is based on its participation in and deep commitment to the open source software community. The development and use of open source has expanded at an exponential rate in recent years.[1] Open source software is now ubiquitous, touching the lives of the millions of Americans who do web searching, email, online shopping, banking, and many other everyday activities. It provides the technological backbone of many large corporations and is critical to the technology operations of the U.S. and many state governments. It is playing an important role in economic development across the globe. Even so, its nature and significance are still not widely understood.

[1] Open source software is also known by the earlier term "free software." "Free" here refers to the freedom to study, modify and share software, rather than availability at no cost; all open source licenses permit and facilitate commercial distribution. The term "free" has the benefit of emphasizing the ethical foundations of open source but can cause confusion if misunderstood to mean gratis. Therefore we have used the label "open source software" in this brief.

The open source model produces software through a mechanism of collaborative development that fundamentally relies on communication of ideas by large numbers of individuals and companies. To understand this model, it is helpful to understand how software is made. Software begins as plain text "source code." Programmers write and edit source code in human-readable programming languages that allow specification of software features and behavior at a high level of abstraction. Software is commonly distributed in machine-executable "object code" form, produced by "compiling" the source code of the software.[2] Since object code consists of unintelligible strings of 1s and 0s, software is effectively unmodifiable unless one has access to its source code.

A good example of an open source project is the Linux operating system kernel, which is one of the most commercially-important open source programs and which is a core component of Red Hat's flagship product, Red Hat Enterprise Linux.[3] The Linux kernel contains several million lines of source code. A

[2] In some cases, source code is not compiled but is run directly by an "interpreter" program. The main disadvantage of interpreted code is poor execution speed.

[3] There are many thousands of open source software packages in wide use. Other well-known examples include the Apache web server, the Firefox web browser, the MySQL database management system, and the GCC compiler collection. A leading web site devoted to open source development, SourceForge.net, now has more than 170,000 registered open source projects.

worldwide community of hundreds of contributors, including many employees of Red Hat, collaborate via the Internet in developing and improving the Linux kernel.

Open source uses a combination of technological and legal means to facilitate collaborative development and commercial exploitation. Typically, an open source package originates as a community-based project that makes its software publicly available in source code form, under licensing terms that grant very broad, royalty-free copyright permissions allowing further use, copying, modification and distribution. The Linux kernel, for example, is licensed as a whole under the GNU General Public License, version 2, the most widely-used open source license. In making source code available and conferring broad copyright permissions, open source differs significantly from traditional proprietary software. A vendor of proprietary software generally develops the software entirely in-house and provides only object code to the user under severely restrictive licenses that allow no rights to copy, modify or redistribute that code. Such vendors retain the source code as a trade secret.

The open source development model has proven to be highly effective in producing software of superior quality. Because there are many developers working as collaborators, innovation happens rapidly.[4] Because of the many who

[4] *See, e.g.*, E. von Hippel, Democratizing Innovation, ch. 7 (2005).

volunteer their time, and the availability of the source code under royalty-free licenses granting generous modification and distribution rights, the cost of producing and improving software is low. Software bugs and security problems are quickly identified and remedied. Moreover, because users have access to the source code, those users can diagnose problems and customize the software to suit their particular needs.

The open source development model originated in the early 1980s. From that time to the present, open source software has been in a constant state of innovation. Software patents, however, have not in any way promoted the innovations of open source. At the time when software was first released under open source licenses, software patents were relatively few in number and case law appeared to limit their availability. *See Diamond v. Diehr*, 450 U.S. 175, 185–86 (1981). By contrast, it was settled that copyright law covered software. Thus the early innovators of open source software had no reason even to consider obtaining patents on their work. Moreover, since at least the early 1990s open source developers have been broadly united in their opposition to the patentability of software.

This widespread opposition is not surprising, because the open, collaborative activity at the heart of open source is fundamentally at odds with the patent system. Patents exclude the public from making, using, or selling patented inventions. An open source developer seeks to contribute code to the community—not to exclude others from using the code. The exclusionary objectives of the patent system are inherently in conflict with the collaborative objectives of open source.

This conflict is more than theoretical. Open source software developers constantly face the hazard that the original code they have written in good faith might be deemed to infringe an existing software patent. It is impossible for a developer to rule out this possibility, because there are now more than 200,000 software patents, and those patents cannot possibly be searched and cleared at reasonable cost. Because of the abstract nature of software patents, determining whether even a single software patent claim is infringed is particularly difficult, even for experts in computer science, and experts often disagree. *See*, *e.g.*, J. Bessen and M. Meurer, Patent Failure 201–03 (2008). The complexity of software projects (open source and otherwise) is such that a single computer program is likely to implement numerous forms of functionality that could possibly be deemed to infringe large numbers of unknown patents. Since code may infringe any number of patents, there is always some possibility of a patent lawsuit that could cost millions of dollars in attorneys' fees and that could result in court orders that effectively nullify the broad grant of rights in open source licenses.

In short, the patent system is not the source of innovation in open source software. Because the system does not reward open source innovation and creates litigation risks for the innovators, the system can only hinder innovation. Thus innovation in open source software continues in spite of—not because of—the patent system. The successes that have been built on the open source model are likely to continue. It is, however, an opportune time to address the standards that govern the subject matter limitations on patentability, because clarification of those standards will unquestionably influence the future of open source software, and the future of the software industry generally. It may be that clarification of those standards will benefit open source by reducing the risk of lawsuits and encouraging greater participation in the open source community, with associated benefits for the economy and society as a whole.[5]

[5] Red Hat participates in the patent system in order to address the risk of patent litigation. It has built a portfolio of software patents, but that portfolio is designed to be used only for the purpose of defending against patent aggression. In keeping with its commitment to protect and advance the open source community, Red Hat has extended a public Patent Promise under which it pledges not to enforce its patents against parties who infringe those patents through their use of software covered by designated open source licenses.

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTE

OTHER AUTHORITIES

ARGUMENT

In inviting briefing on the reach of 35 U.S.C. § 101, this Court has recognized an issue of enormous significance. The resolution of this issue is likely to affect, among other enterprises, open source software. Red Hat therefore welcomes the opportunity to offer its perspective on three of the issues on which the Court has requested briefing: (1) whether subject matter is patent-eligible when it constitutes an abstract idea or mental process; (2) whether a method or process must result in a physical transformation or be tied to a machine to be patent-eligible subject matter under Section 101; and (3) whether the Court should reconsider *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998), *cert. denied*, 525 U.S. 1093 (1999), and *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352 (Fed. Cir. 1999), *cert. denied*, 528 U.S. 946 (1999).

In summary, we contend in Part I that abstract ideas are not patentable when they involve no substantial physical transformation. In Part II, we explain that insubstantial physical transformations, such as running a software-implemented algorithm on a computer, should be deemed insufficient to come within Section 101. Finally, in Part III, we argue that the standard for patentable subject matter in *State Street Bank* and *AT&T* is inconsistent with binding Supreme Court precedents and should therefore be reconsidered and modified.

I. ABSTRACT IDEAS ARE NOT WITHIN THE LIMITS SET BY THE PATENT ACT FOR PATENTABLE SUBJECT MATTER

The language of the Patent Act sets clear limits on patentable subject matter. To be patentable, a claimed invention must belong to one of the four categories of subject matter specified in Section 101—a process, machine, manufacture, or composition of matter. 35 U.S.C. § 101. "If a claim covers material not found in any of the four statutory categories, that claim falls outside the plainly expressed scope of § 101 even if the subject matter is otherwise new and useful." *In re Nuijten*, 500 F.3d 1346, 1354 (Fed. Cir. 2007). In addition, the Supreme Court has held that patent protection may not be granted for "laws of nature, natural phenomena, and abstract ideas." *Diamond v. Diehr*, 450 U.S. 175, 185 (1981).

Diehr is the Supreme Court's most recent statement on the scope of patentable subject matter under Section 101, and it is therefore worth particularly close examination. The case has frequently been misread as a departure from prior Section 101 cases and as a basis for patenting subject matter that is abstract and intangible. In fact, however, *Diehr* reinforces the importance of the Section 101 categories and confirms that intangible subject matter may not be patented, whether directly or indirectly through artful claim drafting.

The question presented in *Diehr* was whether a method for curing raw synthetic rubber using a mathematical algorithm implemented as a computer program was a process within the coverage of Section 101. The Court described a patentable process as "an act, or a series of acts, performed upon the subject-matter to be transformed and reduced to a different state or thing." 450 U.S. at 183 (quoting *Cochrane v. Deener*, 94 U.S. 780, 788 (1877)). "Transformation and reduction of an article 'to a different state or thing' is the clue to the patentability of a process claim that does not include particular machines." *Id.* at 184 (quoting *Cochrane*).

The *Diehr* Court found that the patent before it disclosed steps for transforming raw rubber into a different state (that is, cured rubber), and so it fell within the boundaries of Section 101. 450 U.S. at 184–85. *Diehr* did not suggest a broadening of the traditional requirements for patentable subject matter, but instead restated the requirement that a process patent involve a physical transformation. *Id.* at 183–84.

The *Diehr* Court took care to emphasize the continuing validity of its prior cases barring patents for abstract ideas. It reaffirmed its holding in *Gottschalk v. Benson*, 409 U.S. 63 (1972), that algorithms, or procedures for solving mathematical problems—the building blocks of computer programs—cannot be patented. *Id.* at 186. Likewise, it reaffirmed its holding in *Parker v. Flook*, 437 U.S. 584 (1984), that an algorithm for computing a number that served as an alarm limit was not patentable. It contrasted these intangible processes with the highly tangible subject matter of the *Diehr* patent—curing of rubber. *Id.* at 187–88.

In short, *Diehr* is not properly understood as modifying the traditional requirements for patentability. Instead, *Diehr* reaffirms that abstract ideas by themselves are unpatentable, and that only inventions that are sufficiently tangible are patentable.

II. AN ABSTRACT IDEA DOES NOT BECOME PATENTABLE MERELY BY JOINING IT WITH A COMPUTER

Distinguishing an unpatentable abstract idea from a patentable process or machine is not always straightforward. As *Diehr* recognized, an algorithm that is plainly unpatentable by itself may be a part of a process that is patentable when it involves a physical transformation of the sort that has traditionally been considered patentable, such as an industrial process for curing rubber. 450 U.S. at 184–88. *See also Parker v. Flook*, 437 U.S. at 589–94. The Supreme Court has also recognized that an otherwise abstract idea may be patentable when "tied to a particular apparatus." *Flook*, 437 U.S. at 588, n.9. This Court, in *In re Comiskey*, broadly summarized this case law by saying "a process claim reciting an algorithm [can] state statutory subject matter if it: (1) is tied to a machine or (2) creates or involves a composition of matter or manufacture." 499 F.3d 1365, 1377 (Fed. Cir. 2007).

It is possible, however, to read such limiting language so that it renders abstract ideas patentable. For example, "tied to a machine" could, if read literally without a view to prior case law, be taken to mean "run on a computer." Indeed, *State Street Bank* and *AT&T* adopt this approach. In addressing the scope of Section 101, this Court should now reject this approach, because it is plainly inconsistent with the Supreme Court's holdings in *Benson* and *Diehr*.

In *Benson*, the patent application covered "a method of programming a general-purpose digital computer to convert signals from binary-coded decimal to pure binary form." 409 U.S. at 65. The Court found that the procedure at issue was a mathematical algorithm, and amounted to an unpatentable abstract idea. *Id.* at 65–66. It is important to note, however, that the algorithm had "no substantial practical application except in connection with a digital computer." *Id.* at 71. The claims were intended to "cover any use of the claimed method in a general-purpose digital computer of any type." *Id.* at 64.

Thus the algorithm claimed in *Benson* could have been viewed as "tied to a machine," inasmuch as it was functionally tied to a digital computer. Nevertheless, the Supreme Court held that the claims were abstract ideas outside the scope of Section 101. Thus *Benson* precludes an interpretation of Section 101 that views abstract ideas as patentable based on their implementation in software running on a general purpose computer.

The more recent decision in *Diehr* is consistent with this understanding. The rubber curing process in *Diehr* involved an algorithm and a computer, but the Supreme Court's analysis of subject matter turned on the claim as a whole, which concerned the physical transformation of the rubber—not the implementation of the algorithm in a computer program. 450 U.S. at 184–85. *Diehr* explained that use of the computer did not render the process unpatentable, but the decision makes clear, by its focus on the transformation of rubber, that use of the computer alone does not suffice to make the process patentable. *See id.* at 187.

The *Diehr* Court cautioned against allowing the prohibition on patenting of abstract formulas to "be circumvented by attempting to limit the use of the formula to a particular technological environment." 450 U.S. at 191. "To hold otherwise would allow a competent draftsman to evade the recognized limitations on the type of subject matter eligible for patent protection." *Id.* at 192. In view of this caution, this Court should make clear that the test for patentable subject matter cannot be satisfied by the mere drafting device of joining an otherwise unpatentable algorithm to a general-purpose computer.

It is true that the line between tangible and intangible subject matter is not precise. There is a sense in which a purely mental process that is implemented entirely within the human brain involves a physical transformation, inasmuch as it involves physical neurotransmitter chemicals acting on physical cells. However, the Supreme Court's decisions in *Benson*, *Flook*, and *Diehr* make clear that mental processes are not, by themselves, patentable, because they are too abstract and intangible. Similarly, transitory changes in electronic states that occur in circuits could be characterized as tangible at some level, but these cases show that more is required for Section 101. *Cf. Nuijten*, 500 F.3d at 1353 (Section 101 does not permit patenting claims directed to "physical but transitory forms of signal transmission such as radio broadcasts, electrical signals through a wire, and light pulses through fiber-optic cable"). Thus this Court should make clear that merely joining an algorithm to a general purpose computer is not sufficient to create patentable subject matter.

III. THE STANDARD SET FORTH IN *STATE STREET BANK* AND *AT&T* IS INCONSISTENT WITH THE SUPREME COURT'S CASES

The Federal Circuit has not consistently followed the path laid out in *Diehr* and earlier cases with respect to Section 101. In *State Street Bank*, this Court adopted a new test for patentable subject matter that was not well grounded in the statutory language of Section 101 or the guiding Supreme Court precedents. This Court said the question of statutory subject matter should not focus on the four categories of Section 101. Instead, the Court set forth a new focus: "practical utility." 149 F.3d at 1375. The Court acknowledged that the Supreme Court had held that mere abstract ideas were not be patentable. Nevertheless, it went on to determine that abstract ideas were patentable if they produced "a useful, concrete and tangible result." *Id.* at 1373.

This Court applied the approach of *State Street Bank* in *AT&T*, which concerned a patent for a primary interexchange carrier ("PIC") indicator to be used to calculate billing for long distance telephone calls. The Court found that the claim was a "process" within the meaning of Section 101. Although the Court acknowledged that "the PIC indicator value is derived using a simple mathematical principle," it stressed that it provided "a useful, non-abstract result that facilitates differential billing." 172 F.3d at 1358. The *AT&T* Court misread the *Diehr* case and concluded that Section 101 did not require any physical transformation. *Id.* at 1358–59.

The error in *State Street Bank* and *AT&T* is evident from comparison to the Supreme Court's decision in *Flook*. There, the Supreme Court determined that a method for determining an alarm limit during a catalytic conversion process was not patentable subject matter. The *Flook* Court acknowledged that the application of the method was "useful," 437 U.S. at 585, and there is no doubt it would have satisfied the *State Street Bank* test of "a useful, concrete and tangible result." Nevertheless, the Supreme Court held that the method failed to come within one of the four categories of Section 101. *Id.* at 594.

Likewise, the result in *Benson* is inconsistent with this court's "useful, concrete and tangible result" test. *Benson* concerned an algorithm to be implemented on a digital computer for converting from binary-coded decimal to pure binary form. Plainly, this algorithm was "useful." There were, and still are, immediate commercial and industrial applications of such an algorithm. Indeed, the Court recognized that the *Benson* algorithm had "substantial practical application . . . in connection with a digital computer." 409 U.S. at 71. Moreover, the output of the *Benson* algorithm is a number that is encoded in a particular way. This Court has said that the "useful, concrete and tangible result test" is satisfied "even if the useful result is expressed in numbers." *State Street Bank*, 149 F.3d at 1375. However, the *Benson* Court held that the algorithm was not within the scope of patentable subject matter under Section 101. 409 U.S. at 73.

It is also worth noting that three justices of the Supreme Court have criticized the "useful, concrete and tangible result" test. In *Laboratory Corporation of America Holdings v. Metabolite Laboratories, Inc.*, 548 U.S. 124, 126 S. Ct. 2921 (2006), Justice Breyer dissented from the Court's dismissal of the writ of certiorari as improvidently granted, in an opinion joined by Justices Stevens and Souter. Justice Breyer's opinion took care to point out, in connection with the useful, concrete and tangible result test, that "this Court has never made such a statement." *Id.* at 2928. The dissenting justices further noted that the statement "would cover instances where this Court has held to the contrary." *Id.* (discussing *Benson*, *Flook*, and *O'Reilly v. Morse*, 56 U.S. (15 How.) 62 (1853)).

In *Comiskey*, a panel of this Court recently reappraised the assessment of patentable subject matter under Section 101. The *Comiskey* Court emphasized that abstract ideas are not themselves patentable, even if they have practical utility. 499 F.3d at 1377. *Comiskey* correctly refocuses the Section 101 inquiry. It does not, however, by its terms reject *State Street Bank*, and it states that an unpatentable mental process may become patentable by combining it with a machine. *Id.* For the reasons explained in Part II of this brief, such a statement can easily be misinterpreted so as to undermine the prohibition on patenting abstract ideas. This Court should correct the erroneous standard of *State Street Bank* and *AT&T* by rejecting the usefulness approach and making clear that abstract ideas, even when combined with general purpose computers, are not generally patentable.

CONCLUSION

In view of the requirements of Section 101 and the governing Supreme Court precedents, this Court should reaffirm the prohibition on patenting abstract ideas. It should reject the test set forth in *State Street Bank* and *AT&T* and adopt a test which is consistent with *Diehr* and other Supreme Court case law and which requires an element of physical transformation.

Respectfully submitted,

Richard E. Fontana
Robert H. Tiller
Red Hat, Inc.
1801 Varsity Drive
Raleigh, NC 27606
(919) 754-4847
Attorneys for *Amicus Curiae*
Red Hat, Inc.

April 7, 2008

CERTIFICATE OF FILING AND SERVICE

I hereby certify that on this 7th day of April, 2008, two bound copies of the Brief of *Amicus Curiae* Red Hat, Inc. in Support of Appellee were served, via U.S. Mail, postage prepaid, upon the following:

David C. Hanson
THE WEBB LAW FIRM
700 Koppers Building
436 Seventh Avenue
Pittsburgh, Pennsylvania 15219
(412) 471-8815

Counsel for Appellants

James Toupin
Stephen Walsh
Thomas W. Krause
Raymond T. Chen
OFFICE OF THE SOLICITOR
Post Office Box 15667
Arlington, Virginia 22215
(571) 272-9035

Counsel for Appellees

I further certify that on this 7th day of April, 2008, an original and thirty copies of the Brief of *Amicus Curiae* Red Hat, Inc. in Support of Appellee were hand filed at the Office of the Clerk, United States Court of Appeals for the Federal Circuit.

The necessary filing and service were performed in accordance with the instructions given me by counsel in this case.

THE LEX GROUP^{DC}
1750 K Street, N.W., Suite 475
Washington, DC 20006
(202) 955-0001

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE REQUIREMENTS

1. This brief complies with the type-volume limitation of Federal Rule of Appellate Procedure 32(a)(7)(B). The brief contains 2,630 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(a)(7)(B)(iii).

2. This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6). Prior to submission to counsel's printing service, the brief was prepared by counsel in a proportionally spaced typeface using OpenOffice.org Writer 2.3 in Nimbus Roman No9 L 14 point font.

Richard E. Fontana
Robert H. Tiller
Attorneys for *Amicus Curiae*
Red Hat, Inc.

April 7, 2008