Loyola Consumer Law Review

Volume 11 | Issue 2 Article 5

1999

Protecting Intellectual Property Rights in Software: The Software Patent

Raj Sardesai

Michael J. Ram

Protecting Intellectual Property Rights in Software: The Software Patent

by Raj Sardesai
and Michael J. Ram

IBM is launching a software patent war. The company holds more patents than any other company in the world and is now looking into filing patent-infringement suits against well-known names like Oracle, Computer Associates, Adobe Systems, Intuit, and Informix. Insiders say IBM could collect $1 billion a year from patents.[1]

Hardware, and related software are now a multi-billion dollar a year industry. The surge in creativity in the software industry and the accompanying growth in software market was brought on by two important developments in the computer industry - the Internet and the high-performance processors, plentiful memory capacities, and rich multimedia capabilities of today's most powerful home and office personal computers.[2] Therefore, the ability of high-tech companies to patent software and computer programs significantly affect the consumer markets, choice among products, and the prices for products. The limited monopoly that a patent offers its holder in effect encourages companies to invest substantial sums of money on new projects with the prospect of recooping costs and expenses through patents and licenses.

Today, even the lowest-priced computers are amazingly powerful, with advanced capabilities to process sound, graphics, data, and digital communication channels. With software that takes ingenious advantage of these powers, desktop-PC owners can now do things that weren't possible just a few years ago. The available software titles range from business applications to kids' educational programs to engineering and technical applications to entertainment programs. The size of the market in dollars is phenomenal. In the household market segment alone, the 1996 fourth-quarter sales, which include the holiday season, surpassed $10 billion.[3] The growth in the software industry is expected to continue, with newer titles integrating with the Internet to benefit from the vast "information pool" available on the Internet.

Many new software products already provide seamless connectivity with the Internet. Seamless connectivity relates to the ability of a software program to directly access the Internet without first having to exit the immediate program. For example, Quicken, which began a decade ago as an automated checkbook, has in its latest versions (which include Quicken Deluxe 98 and Quicken Home and Business 98) as complete financial planning, tracking, and analysis packages. Using seamless connectivity with the Internet, the new Quicken provides constantly updated stock and mutual fund quotes, portfolio updates, credit-card transactions, and news feeds.[4] Two important issues for continued growth of this industry are regulating the Internet and protecting the intellectual property rights in software. Others have described the Internet and explored the debate over Internet regulation.[5] This article focuses on protecting the intellectual property rights in software, which directly affects the prices and availability of new products in the large consumer market for computers and computer-related products.

Significance of Patent Rights

Patent protection gives the inventor the right to exclude others from making, using, or selling the patented art. The Supreme Court has excluded from the scope of patent protection certain fundamental building blocks of science such as laws of nature, scientific principles, and mathematical algorithms on the grounds that they are "too important to be subjected to private control."[6] The U.S. Patent and Trademark Office, ("PTO"), initially took the view that the computer programs were not patentable because they could be characterized as sequences of mental steps and/or mathematical algorithms.[7] CONTU, a commission established by the United States Congress, also recommended that the intellectual work embodied in computer software should be protected principally under copyright law.[8] Thus, the literary form of computer software rather than its utilitarian substance was thought to be the better guide for determining the form of legal protection. However, in a majority of software titles, the utilitarian substance and the ideas embodied therein, rather than the literary form, constitutes the heart of the software and needs the most protection. Also, copyright law is not fully effective in protecting the software as copyrights protect only the physical manifestation and not the ideas embodied therein. Only patents protect ideas.

The Supreme Court's decision in *Diamond v. Diehr* allowed, for the first time, patent protection for computer programs.[9] Since this decision, the PTO has been issuing software-related patents in increasing numbers. In 1996, the PTO issued Examination Guidelines for Computer-Related Inventions.[10] The PTO further supplemented these guidelines on October 20, 1998. The guidelines are designed to "assist patent examiners in reviewing applications during patent prosecution and are intended to be consistent with Supreme Court and Federal Circuit precedent."[11] This article examines the patentability of software in the light of the above guidelines and the case law controlling the patentability issue.

Law Before The PTO Guidelines

As mentioned above, the Supreme Court and the United States Circuit Courts of Appeals, before 1981, had frequently held computer programs and software unpatentable. Their main concern was preventing patents and limited monopolies on products or ideas that were nothing more than a physical embodiment of an abstract idea, mathematical algorithm or law of nature. For example, in *Gottschalk v. Benson*, the Supreme Court rejected an attempt to patent "a method of programming a general-purpose digital computer to convert signals from binary-coded decimal form into pure binary form."[12] The Court held that allowing a patent on a computer program that simply converts one set of numbers into another would preempt the mathematical formula on which the software was based and "in practical effect would be a patent on the algorithm itself."[13]

In 1978, the Supreme Court again denied a patent covering software in *Parker v. Flook*.[14] Flook's software was designed to control catalytic conversion of hydrocarbons. A catalytic conversion process requires control of process variables such as temperature and pressure within a narrow range for safety and conversion efficiency. In the claimed process, when the process variables went outside the allowable range, an alarm alerted the operator to unfavorable operating conditions. The alarm limits were set as variations of the base, tolerable alarm. Flook's patent application covered a method of updating the alarm limits. The method consisted mainly of:

1. Measuring the present value of the process variable (e.g. pressure or temperature);
2. Using an algorithm to calculate an updated value of the alarm limit; and
3. Resetting the actual alarm limit to the updated value calculated in Step 2.

The Supreme Court held that the only novel feature of the *Flook* method was the mathematical formula and hence, the claim was nothing more than an attempt to a patent a mathematical formula.[15] Flook argued that the adjustment of alarms after solving the mathematical equations resulted in a post-solution activity, rendering his process patentable. The Supreme Court rejected that argument:

> [t]he notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance. A competent draftsman could attach some form of post-solution activity to almost any mathematical formula; the Pythagorean theorem would not have been patentable, or partially patentable, because a patent application contained a final step indicating that the formula, when solved, could be usefully

applied to existing surveying techniques.[16]

The Court also found that the process was unpatentable, not because it contained a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention.[17]

The Supreme Court Signals Change

In 1981, the Supreme Court in *Diamond v. Diehr* held that software designed to control a rubber curing process was patentable.[18] This critical Supreme Court decision made patent protection a possibility for software and other computer programs. The patent at issue in *Diehr* involved software for programming a temperature decay equation well known in the chemical industry. Using real-time temperature measurements in a process control circuit, the computer calculated the time needed to cure the rubber and signaled the precise moment to open the mold. The process required experimentally derived values, namely an activation energy constant unique to each batch of the compound being molded and a constant dependent upon the geometry of the particular mold used in the process. The invention also provided for initiation of an interval timer upon the closure of the mold for monitoring the time elapsed after the closure.

The Court found that the "claims" (the section of the patent application where the applicant specifies in exact words what is sought to be patented) sought protection for a *process* of curing synthetic rubber; therefore, the claimed invention was patentable subject matter.[19] The Court observed that the claims "seek only to foreclose others the use of that [Arrhenius] equation in conjunction with all of the other steps in their claimed process;" therefore, what was claimed was not just a mathematical algorithm but rather an actual chemical process embodying that algorithm.[20]

Analyzing *Flook* After *Diehr*'s Teachings

The alleged novelty in the *Flook* patent application was the use of a mathematical formula to set alarm limits in the catalytic conversion process. In *Diehr*, a formula was used to control a rubber curing process by giving a signal to open the mold. The *Diehr* specification contained a full disclosure not only of the principle or formula involved, but also how to apply the formula to the curing process. In *Flook*, the applicant sought broader protection by claiming a method that could potentially be used against any device that performs the steps, regardless of the actual components employed. Hence, the downfall in *Flook* occurred by not specifying actual alarm or process components; in effect, the applicant claimed nothing more than a mathematical algorithm. The application in *Flook* thus only taught a formula, and not how to apply it to the catalytic conversion process. Also, a human operator was required to make important decisions that will determine the efficacy of the formula such as selecting the appropriate margin of safety and the weighing factor. The important lesson to be learned from *Diehr* and *Flook* is that software patents should be narrowly drawn and the formulae, including all constants and multipliers which form the basis of the software, should be fully specified in the specification of the patent application so that a person skilled in the art can use the teachings without extraneous input.

In *Diehr*, Justice Rehnquist, writing for the majority, offered the following guidance as to what distinguishes a patentable process:

> 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.[21]

In *Flook*, the software determined the alarm limit, a numerical value which alone does not in any way alter or control a physical property, or transform or reduce an article to a different state or thing. It must be applied to the process by an operator for the value derived to be part of the process that it was the intention to protect. Based on the later teachings of the Court in *Diehr*, the subject matter in the *Flook* claims would be patentable if it had been tied to the calculation of the alarm limit to a step that controls the catalytic conversion process, resulting in a different result through such an application.

Recent Trends

In 1994, the Federal Circuit Court of Appeals decided *In re Alappat, en banc*, with eleven judges taking part in the decision.[22] Since the Federal Circuit Court of Appeals is a special court established to hear all appeals in case involving patents, this was and continues to be an important decision on the issue of software patentability.

The invention claimed in *Alappat* overcomes the problem of aliasing that occurs when a display of a waveform on a computer screen rises or falls rapidly. There are a finite number of pixels on a screen and these are arranged in a rectangular grid form. A waveform is displayed as a series of illuminated pixels. When the waveform rises or falls slowly, the human eye sees it as a continuous line. However, when the waveform rises or falls rapidly, the eye sees it as discontinuous and jagged. This aliasing was eliminated by the *Alappat* software, which varied the intensity of pixels in the proximity of the waveform in such a way that a rapidly moving waveform looked continuous.

The Federal Circuit majority found that the Patent Office had erred when it denied a patent to Alappat by holding that the claim was nothing more than an attempt to wholly preempt a mathematical algorithm for computing pixel information.[23] The court also rejected the view that the invention was unpatentable because it fell within the "judicially created mathematical algorithm" exception to 35 U.S.C § 101.[24] Citing *Diehr*, *Flook*, and *Benson*, the court clarified that the Supreme

Court never intended to create an overly broad, fourth category of subject matter excluded from patent protection.[25] The claims in the *Alappat* patent read on a general-purpose computer. The Federal Court majority did not view that as a bar to patentability. The court held that "a general purpose computer in effect becomes a special-purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."[26] The court further noted that "such programming creates a new machine."[27] Judge Rich noted that a general-purpose computer executing Program A is a "different machine" than the same computer executing Program B.[28] Additionally, the court considered the step of "outputting illumination density data" as being sufficient to satisfy the physical act requirement alluded to in *Flook* and *Diehr*.[29]

Distinguishing *Alappat* and *Flook*

The computer software output in the inventions involved in both *Alappat* and *Flook* were numerical values. However, the *Alappat* claims were to a machine while *Flook* claimed a method. Nonetheless, the description of the method in *Flook* could be modified to more closely resemble the machine in *Alappat*. Alappat claimed the use of the numbers to update the intensity of pixels on the computer screen. Flook required an operator to use the numbers to update the alarm limits. Following the reasoning in *Alappat*, the subject matter in the *Flook* claims could be statutory *and therefore patentable* if his claims were to a machine which applied the values derived by the formula to change the operation of the machine. The claim in *Flook* could be rewritten to cover a machine by adding the following preamble in the patent application:

> An alarm generating machine for receiving process data representing measured process variables, analyzing the process data, comparing said processed data with predetermined safe operating conditions, establishing alarm limits and displaying said alarm limits on a display means comprising.

The claim could then have several means plus function elements comprising the invention. Additionally, or in the alternative, the claim could specify machine controlled action if the alarm limits are exceeded.

Law After *Alappat* But Before The PTO Guidelines

After *Alappat*, the Federal Circuit Court of Appeals decided three important cases - *In re Warmerdam, In re Lowry* and *In re Trovato*.[30] In *Warmerdam*, the claims were to a method and a machine for controlling the motion of objects, such as robots, to avoid collisions with other moving or stationary objects. The Federal Circuit held that Warmerdam's method claims were not directed to statutory subject matter because they were no more than an "abstract idea."[31] The court referred to the necessity of "transformation or reduction of subject matter."[32] Warmerdam's method claims did not mention a computer or signals, but described an algorithm to generate data. Interestingly, the court observed that *Warmerdam*'s allowed machine claims, although not at issue in the appeal, were "clearly patentable subject matter" and noted that "the storage data in a memory physically alters the memory, and thus, in some sense, gives rise to a new memory."[33] The reasoning is similar to that in *Alappat* where the Federal Circuit majority held that programming transforms a general-purpose computer into a new machine.[34]

In *Lowry*, the subject matter was an object-oriented data structure.[35] The court held the subject matter patentable.[36] Noting that data structures are more than mere abstractions, the court observed that they are physical entities that provide increased efficiency in computer operation.[37] The court also observed that the data structures are specific electrical or magnetic structural elements in a memory and provide "tangible benefits: data stored in accordance with the claimed data structures are more easily assessed, stored and erased."[38] There is a common thread in the court's reasoning in *Alappat*, *Warmerdam* and *Lowry*, namely, a general-purpose computer transforms into a "new machine" when it is programmed or a data structure is introduced into its memory.

The *Trovato* claims were to a method and apparatus that uses linear-programming techniques to determine the least cost or optimal path between two locations. The Federal Circuit first held that the subject matter was nonpatentable, noting that "Trovato claims nothing more than the process of performing a numerical calculation."[39] Trovato then petitioned successfully for an *en banc* rehearing. The court vacated the first judgment, withdrew the opinion accompanying the judgment and remanded the case to the PTO "for consideration in light of *Alappat* and any new guidelines adopted by the Patent and Trademark Office for examination of computer-implemented inventions."[40]
The court noted that "consistent with *Alappat*, the proposed guidelines direct patent examiners to apply all of the requirements of Title 35 when examining applications claiming computer software instead of rejecting such applications under section 101."[41]

PTO Guidelines

In 1996, the PTO released guidelines for software-related inventions.[42] The guidelines provide a set of substantive standards for addressing the statutory subject matter issue and describe a series of procedural steps to be used when reviewing applications computer-implemented inventions. The guidelines attempt to eliminate the "form over substance" distinction that existed in inventions claimed as the machine embodiment of a process. When

a claim is found to encompass any

and every product embodiment of the underlying process, and if the underlying process is statutory, the product claim should be classified as a statutory product. By the same token, if the underlying process invention is found to be non-statutory, office personnel should classify the 'product' claim as a non-statutory product.[43]

Non-statutory subject matter (i.e., abstract ideas, laws of nature and natural phenomena) do not become statutory merely through a different form of claim presentation.

The guidelines also identify examples of "safe harbors" or subject matter that is either clearly statutory or non-statutory. A claimed process is statutory if it results in a physical transformation outside the computer, *i.e.* falls into one or both of the following specific categories:

Independent Physical Acts (Post-Computer Process Activity)

A process is statutory if it requires physical acts to be performed outside the computer independent of and following the steps to be performed by a programmed computer, where those acts involve the manipulation of tangible physical objects and result in the object having a different physical attribute or structure. Thus, if a process claim includes one or more post-computer process steps that result in a physical transformation outside the computer, the claim is statutory.[44]

Examples of this type of statutory process include:

- A method of curing rubber in a mold which relies upon updating process parameters, using a computer processor to determine a time period for curing the rubber, using the computer processor to determine when the time period has been reached in the curing process and then opening the mold at that stage.[45]
- A method of controlling a mechanical robot which relies upon storing, in a computer, data that represents various types of mechanical movements of the robot, using a computer processor to calculate positioning of the robot in relation to given tasks to be performed by the robot, and controlling the robot's movement and position based on the calculated position.[46]

Manipulation of Data Representing Physical Objects or Activities (Pre-Computer Process Activity)

A second statutory process is one that requires the measurements of physical objects or activities to be transformed outside of the computer into computer data, where the data comprises signals corresponding to physical objects or activities external to the computer system, and where the process causes a physical transformation of the signals which are intangible representations of the physical objects or activities.[47]

Examples of this type of claimed

statutory process include:

- A method of using a computer processor to analyze electrical signals and data representative of human cardiac activity by converting the signals to time segments, applying the time segments in reverse order to a high pass filter means, using the computer processor to determine the amplitude of the high pass filter's output, and using the computer processor to compare the value to a predetermined value. In this example, the data is an intangible representation of physical activity, *i.e.* human cardiac activity. The transformation occurs when heart activity is measured and an electrical signal is produced. This process has real world value in predicting vulnerability to ventricular tachycardia immediately after a heart attack.[48]

- A method of using a computer processor to receive data representing Computerized Axial Tomography ("CAT") scan images of a patient, performing a calculation to determine the difference between a local value of a data point and an average value of the data in a region surrounding the point, and displaying the difference as a gray scale for each point in the image, and displaying the resulting image. In this example the data is an intangible representation of a physical object, i.e., portions of the anatomy of a patient. The transformation occurs when the condition of the human body is measured with X-rays and the X-rays are converted into electrical digital signals that represent the condition of the human body. The real world value of the invention lies in creating a new CAT scan image of body tissue without the presence of bones.[49]

- A method of using a computer processor to conduct seismic exploration, by imparting spherical seismic energy waves into the earth from a seismic source, generating a plurality of reflected signals in response to the seismic energy waves at a set of receiver positions in an array, and summing the reflection signals to produce a signal simulating the reflection response of the earth to the seismic energy. In this example, the electrical signals processed by the computer represent reflected seismic energy. The transformation occurs by converting the spherical seismic energy waves into electrical signals, which provide a geophysical representation of formations below the surface of the earth. Geophysical exploration of formations below the surface of the earth has real world value.[50]

The guidelines focus on the "useful

art" aspect of 35 U.S.C. § 101. To satisfy the requirements of § 101, an applicant also must show implicitly or explicitly that the claimed invention is "useful" for some purpose. In the examples above, the PTO has explicitly identified the "real world value" of the inventions. Therefore, it is suggested by this author that practitioners explicitly state the "real world value" in the patent application specification.

Further, the guidelines recognize that

> in some instances, certain kinds of post solution acts will not further limit a process claim beyond the performance of the preceding mathematical operation step even if the acts are recited in the body of the claim.[51]

If, however, the claimed acts represent some "significant use" of the solution, those acts will invariably impose an independent limitation on the claim.[52] A "significant use" is any activity that is more than merely outputting the direct result of the mathematical operation.[53] For example, acts that require the conversion of a series of numbers representing values of a wavefunction equation for a chemical compound into values representing an image that conveys information about the three-dimensional structure of the compound and the displaying of the three-dimensional structure cannot be treated as being part of the mathematical operation.[54]

Law after PTO Guidelines

In March 1996, shortly after the PTO Guidelines were issued, a U.S. District Court in Massachusetts decided *State Street Bank & Trust Co. v. Signature Financial*.[55] State Street sought a declaratory judgment that Signature's patent for a computerized accounting system for managing mutual fund investment structure is invalid and unenforceable.[56] State Street alleged that Signature's subject matter is non-statutory because the invention claims a mathematical algorithm.[57] Signature countered that its data processing system is a computer-implemented invention that is patentable under both prior case law and the PTO guidelines.[58]

Signature's invention was claimed in means- plus- function language as an "apparatus." The specification disclosed a general-purpose personal computer programmed with software by which the invention operates.[59]

The court applied a two-part mathematical algorithm/physical transformation test to determine the patentability issue.[60] The court noted that regardless of whether the claim is drafted as a process or apparatus, the Federal Circuit has held that the mathematical algorithm/physical transformation test for statutory subject matter under § 101 applied even to "true apparatus" claims.[61] This two-part test is also known as the Freeman-Walter-Abele test, which the Federal Circuit has recently described as follows:

> It is first determined whether a mathematical algorithm is recited directly or indirectly in the claim. If so, it is next determined

> whether the claimed invention as a whole is no more than the algorithm itself; that is, whether the claim is directed to a mathematical algorithm that is not applied to or limited by physical elements or process steps. Such claims are non-statutory. However, when the mathematical algorithm is applied to one or more elements of an otherwise statutory process claim, the requirements of section 101 are met.[62]

Applying the mathematical algorithm part of the test, the court found that although Signature's patent claims do not directly recite a mathematical formula, the data processing system is an apparatus specifically designed to solve a mathematical problem.[63] Interpreting the claims in light of the specifications, the court held that the claims recite means for solving a series of mathematical problems.[64] The court noted that an invention that inputs, processes, and outputs numbers must, *by definition*, perform mathematical operations.[65]

The court also held that Signature's data processing system fails the physical transformation test.[66] The court reasoned that the system does not "involve the transformation or conversion of subject matter representative of or constituting physical activity or objects."[67] The court concluded by saying,

> [a] change of one set of numbers into another, without more, is insufficient to confer patent protection...[t]he invention does nothing other than present and solve a mathematical algorithm and, therefore, is not patentable.[68]

On appeal, the Federal Circuit reversed and concluded that the Signature patent claims fell within statutory subject matter. The court held that:

> transformation of data, representing discrete dollar amounts by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, or calculation, because it produces 'a useful, concrete and tangible result' - a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades.[69]

The court also observed that the district court erred by applying the Freeman-Walter-Abele test, and noted that the test has little, if any, future applicability to determining the presence of statutory subject matter.[70] "Application of the (Freeman-Walter-Abele) test could be misleading, because a process, machine, manufacture, or composition of matter employing a law of nature, natural phenomenon, or abstract idea is patentable subject matter even though a law of nature, natural phenomenon, or abstract idea would not, by itself, be entitled to such protection."[71] Citing *Alappat*, the court held that "the mere fact that a claimed invention involves inputting numbers, calculating numbers, outputting numbers, and storing numbers, in and of itself, would not render it non-statutory subject matter, unless, of course, its operation does not produce a *"useful, concrete and tangible result."*[72]

Software Patentability Analyzed

Software, by its very nature, performs a mathematical calculation and/or data organization/ manipulation. The determining factor in patentability is the relationship between the claims and physical objects or process steps, the limitations on the claim, and the practical application of the results. Following the steps performed by a programmed computer, if there is physical transformation of an object to a different physical attribute or structure, the claim is statutory. Examples include software to determine the time for curing of rubber and opening of a mold, software to control a robot's movement and position based on a mathematical calculation.

If the input to the software program comprises data representing physical objects or activities external to the computer system, and there is physical transformation of the data into a form suitable for processing by the computer system, then the subject matter is statutory. Examples include software designed to monitor human heart activity and software designed to analyze seismic energy waves where each program transforms the respective information into electrical signals. An expansive view is that any act performed to create data that will then be used in a computer process representing a practical application of one or more mathematical operations is patentable. Such acts could be construed as independent limitation on the claim because they are not dictated by the algorithm but by other limitations which require certain antecedent steps.

A claim, which includes a software function as well as providing a practical application of an abstract idea or mathematical algorithm in the technological arts, is patentable. For example, a computer process that simply calculates a mathematical algorithm that models noise is non-statutory. However, a claimed process for digitally filtering noise employing the mathematical algorithm *is* statutory.

A claim requiring that the direct result of a mathematical operation be evaluated and transformed into something else is a limitation on the claim, making the claim patentable. The results of the mathematical operation are then construed to have been transformed into something of significant use. A "significant use" is any activity, which is more than merely outputting the direct result of the mathematical operation. For example, conversion of a series of numbers representing values of a wavefunction equation for a chemical compound into values representing an image (for display) that conveys information about the three-dimensional structure of the compound constitutes "significant use." Note that, in *Flook*, the post-solution activity of "updating alarm limits" was found not to be a "significant use" of the solution.[73]

An important practice tip is to structure a software claim , wherever possible, in a manner in which the end result is not simply a number but involves an actual physical act. It may not always be possible to have an actual physical act as an end result. However, if the solution is put to some "significant use", the subject matter may become patentable. The patentability will hinge on a clear characterization of the significance of post-solution activity in relation to the invention as a whole and the mathematical operations recited in the claim. Note that form over substance distinctions, such as post-solution activity of "updating alarms" involved in *Flook*, will not save the claim.[74]

Software is also patentable if its functionality is based on input data that characterizes a physical object. The input data could be in any form such as electrical / magnetic signals or numbers. The input may be manual or automatic. The act of measuring a physical object to create data for use in a process, a practical application of mathematical operations, further limits the claim beyond the mathematical operations per se.

If none of the above is applicable to a particular software program, the software is still patentable if it has a "practical application." In applying this test, treat the mathematical formula as if it was a familiar part of the prior art, then evaluate the claim to determine whether some inventive concept with practical application is disclosed, thereby producing a "useful, concrete and tangible result."

Conclusions

Although the intellectual property embodied in software does not easily fit within the traditional patent protection framework, recent developments in the law and the PTO guidelines provide adequate framework to determine the patentability of software. When analyzing whether claims directed to software describe patentable subject matter, courts, examiners, and practitioners must adhere closely to the precedent law on patentability, with the PTO Guidelines providing a "simple-to-use" guide for the analysis.

To determine software patentability, courts have repeatedly applied the two-part Freeman-Walter-Abele test whose second part inquires whether the claimed invention is applied to or limited by physical elements or process steps. If the claimed computer-related invention transforms or reduces subject matter to a different state or thing, it is statutory under Section 101 even if it recites, directly or indirectly, a mathematical algorithm. Consequently, a software interacting with the physical environment external to the computer has a higher likelihood of withstanding statutory subject matter challenges in litigation than other kinds of software which only manipulate data. When an actual physical act as an end result of applying the software is not possible, the software may still be patentable. If the functionality of the software is based on input data, which characterizes a physical object, the software may be patentable. The chances of patentability are enhanced if there is a physical transformation of data, making the data suitable for input to the software.

Additionally, the United States Court of Appeals for the Federal Circuit has recently observed that the Freeman-Walter-Abele test and its requirement that a claim be applied in some manner to physical elements or process steps has little, if any, future applicability in determining the presence of statutory subject matter.[75]

The software is patentable if it has practical application or the results of the calculations done by the software are put to some significant use. Form over substance distinctions such as casting a claim in terms of a statutory category(e.g. apparatus embodiment of software), or post-solution activity that adds nothing new may not be sufficient to save a claim. The specific utility of the invention must be apparent from the claim. The determining factor is whether the software produces "useful, concrete and tangible results."

Statutory subject matter is also comprised of computer-related processes limited to a practical application in the technological arts. A process that merely manipulates an abstract idea or performs a purely mathematical algorithm is non-statutory despite the fact that it might inherently have some usefulness. For such subject matter to be statutory, the claimed process must be limited to a practical application of the abstract idea or mathematical algorithm in the technological arts.

Endnotes

1 Ira Sager, *Big Blue is Out to Collar Software Scoff Laws*, Business Week, Mar. 17, 1997, at 34.

2 *See Technology Buyer's Guide*, Fortune, Nov., 1997, at 188.

3 *See U.S. Households on Software Shopping Spree*, PC Magazine, Jul. 23, 1997.

4 *See Technology Buyer's Guide*, Fortune, p190.

5 *See Regulating the Net: Case Studies in California and Georgia Show How Not to do it*, 9 Loy. Consumer L. Rep. 230 (1997).

6 Paul Golsstein, *Copyright, Patent, Trademark and Related State Doctrines* 525 (2d ed. 1982).

7 *See* Nelson Moskowitz, *The Metamorphosis of Software-Related Inventions Patentability*, 3 Computer L. J. 273 (1982).

8 *See generally, National Commission on New Technological Uses of Copyrighted Works*, Final Report (1979).

9 450 U.S. 175 (1981).

10 Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (1996).

11 *Id.*

12 409 U.S. 63, 71 (1972).

13 *Id.* at 71.

14 437 U.S. 584 (1978).

15 *See id.* at 587.

16 *Id.* at 589.

17 *See id.*

18 *See* Diamond, 450 U.S. at 175.

19 *See id.* at 184 (emphasis added).

20 *See id.* at 187.

21 *See id.* at 184.

22 33 F. 3d 1526 (Fed. Cir. 1994).

23 *See id.* at 1539.

24 *See id.* at 1542.

25 *See id.* at 1543.

26 *Id.* at 1544.

27 *Id.*

28 *See id.* at 1546.

29 *See id.* at 1540.

30 33 F. 3d 1354 (Fed. Cir. 1994), 32 F. 3d 1579 (Fed. Cir. 1994), 42 F. 3d 1376 (Fed. Cir. 1994).

31 *See In re Warmerdam*, 33 F. 3d at 1360.

32 *Id.*

33 *Id.*

34 *See In re Alappat*, 33 F. 3d at 1544

35 *See In re Lowry*, 32 F. 3d at 1579.

36 *See id.*

37 *See id.*

38 *Id.*

39 *In re Trovato*, 42 F. 3d at 1379.

40 *In re Trovato*, 60 F. 3d 807 (Fed. Cir. 1995).

41 *Id.*

42 *See* Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (1996).

43 *Id.* at 7482.

44 *See id.* at 7483.

45 *See id.* at 7484.

46 *See id.*

47 *See id.*

48 *See id.*

49 *See id.*

50 *See id.*

51 *Id.* at 7485.

52 *See id.*

53 *See id.*

54 *See id.*

55 927 F. Supp. 502 (1996).

56 *See id.* at 504.

57 *See id.*

58 *See id.* at 505.

59 *See id.* at 510.

60 *See id.*

61 *See id.*

[62] *Id.* at 513.

[63] *See id.*

[64] *See id.*

[65] *See id.* (emphasis added).

[66] *See id.* at 514.

[67] *Id.*

[68] *Id.*

[69] *Id.* at 1373.

[70] *See id.* at 1374.

[71] *Id.*

[72] *Id.*

[73] *See Parker*, 437 U.S. at 590.

[74] *See id.*

[75] *See* State Street, 149 F. 3d 1374.

CLR