Patents for Software-Related Inventions

Jeffrey R. Kuester
Thomas, Kayden, Horstemeyer & Risley, L.L.P.

Ann K. Moczyunas
Kennesaw State University

Date published: 4 April 2003

[Note: This article was first published in Multimedia and Technology Licensing Reporter, vol. 3, no. 11 (April 1997). Use is by permission of copyright owner Thomson West.]

Abstract: The topic of patents for software-related inventions is a very technical and complicated area. There is an extensive recent history of discussions and heated debates regarding the wisdom of using the patent system to protect software. This is especially true in light of recent developments in copyright law, which suggests that computer software will not find very much protection under the copyright law.


The topic of patents for software-related inventions is a very technical and complicated area. Furthermore, there have recently been many discussions and heated debates regarding the wisdom of using the patent system to protect software. This is especially true in light of recent developments in copyright law, which suggests that computer software will not find very much protection under the copyright law. However, before examining the peculiarities and issues involved with patents for particular areas of technology, certain basic information regarding patent protection should be understood.
Basic Patent Information

The founders of the United States provided in the Constitution that “Congress shall have the power . . . to promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries.” A patent is the grant from the United States government to the owner of an invention to exclude all others from making, using or selling the invention for a limited time. In exchange for the patent rights, the inventor is required to disclose the invention in enough detail to allow other people to use the invention after the monopoly expires and to learn from the invention to build on its ideas. Thus, the patent system was founded to promote technological development by providing incentive to developers of new products through enabling them to prevent others from “stealing their ideas” through profiting without spending the necessary research time and money.

The Patent Act is broad and general in its language describing the proper subject matter for a patent:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

However, not all “inventions” are patentable; “excluded from such patent protection are laws of nature, natural phenomena, and abstract ideas.” “An idea in and of itself is not patentable.”

There are different types of patents. A utility patent is granted for the functional aspect of a machine, an article of manufacture, a composition of matter or a process. A utility patent currently lasts for 17 years from the date of the grant (provided that maintenance fees are paid at certain intervals after the patent issue date: 3 and 1/2 years, 7 and 1/2 years, and 11 and 1/2 years). A design patent is granted for the ornamental appearance of an article, but protects only the non-functional aspects of the design. A plant patent is granted for the discovery and asexual reproduction of a new variety of plant.

Someone who infringes the patent by making, selling, or using the invention without a license may be liable to the patent owner for damages (no less than a reasonable royalty), pre-judgment interest, treble damages (for willful infringement), and attorneys fees and costs in egregious cases. The infringer’s infringing actions can also be stopped by court order.

The Criteria for Granting a Patent

In the United States, a person is entitled to a patent if:

1. the person is the inventor;
2. the invention is the proper subject matter for a patent: machines, articles of manufacture, compositions of matter, and processes;\textsuperscript{13} and
3. the invention is “useful,” “new”\textsuperscript{14}, and “unobvious.”\textsuperscript{15}

For most types of inventions, the first two requirements for patentability normally present very little difficulty.\textsuperscript{16} Generally speaking, an invention is “useful” if the invention has a utilitarian or commercial value; an invention is “new” if the invention is the first\textsuperscript{17} embodiment of the idea in a useful thing or process; and an invention is “unobvious” if the invention would not have been obvious to a person reasonably skilled in the pertinent art, given what already existed in the particular area of the invention (the prior art).\textsuperscript{18} This last requirement is often the most difficult hurdle for a new invention since it is normally a very subjective process to determine whether the differences between the new invention and the prior art are “obvious” solutions to known problems.

\textit{The Patent Application Process}

The process for obtaining a patent on an invention typically starts with a patent attorney preparing and filing a patent application for the inventor and owner of the invention with the United States Patent and Trademark Office (“PTO”).\textsuperscript{19} The first inventor is entitled to the patent over any subsequent inventors, even if the subsequent inventor files first, provided the first inventor can prove the earlier date of invention. The application must be filed no later than one year after:

1. public use of the invention;
2. the placing on the market of the invention for sale in the United States; or
3. the publication of a description of the invention anywhere in the world.\textsuperscript{20}

In exchange for the patent, the inventor is required to disclose the material aspects of the invention, many times including trade secrets. Thus, there is a tension between the inventor’s desire to disclose as little as possible and the patent law’s requirement of disclosure. The application itself must meet very technical requirements for describing the invention. The United States Code provides in part:

\begin{quote}
The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor of carrying out his invention.\textsuperscript{21}
\end{quote}

Because of the specialized nature of a patent application, the applicant should seek the advice of a patent attorney or patent agent having knowledge and/or education in the discipline of the invention. Only those persons admitted to the U.S. Patent Bar are allowed to file and prosecute patent applications for others before the PTO.
What happens when the PTO says no?

A patent application can travel up two levels in the PTO and then two more levels in the federal court system before reaching the “final” word on whether the claimed invention is patentable. In addition, once a patent has been issued, it can be challenged in a lawsuit filed in a United States District Court; a decision from those courts may be appealed to the United States Court of Appeals for the Federal Circuit (its equivalent prior to 1982 was the Court of Customs and Patent Appeals) and then to the United States Supreme Court under certain circumstances.

The courts decide the validity of a patent with regard to both the language of the Patent Act and to interpretations of the Patent Act in other cases. Those other interpretations form the body of case law which, in the U.S. judicial process, is respected as precedence.

Patentability of Computer Software: An Overview

The courts, particularly the appellate courts (Federal Circuit and Supreme Court), have not seen many cases involving issues about the patentability of computer programs. Before even getting to the traditionally most difficult area of “obviousness,” the courts have been struggling with the subject matter requirement for patentability. Specifically, the courts have had trouble deciding whether various types of software can accurately be characterized as falling under any of the areas of patentable subject matter, such as “processes” or “machines.” Further, the advances in computer technology have made the early cases (of only twenty years ago) seem archaic with respect to inventions and common knowledge within the industry.

As of today, inventions that include computer programs as all or part of the invention may be patentable under certain circumstances. However, before discussing recommendations for software owners or the current state of the patent law as applied to software-related inventions, an examination of the history and development of this area of law is in order.
The struggle with the definition of “algorithm”: Gottschalk v. Benson

Amazingly enough, although computer programs have been developed for many decades, the United States Supreme Court did not see a case that raised the issue of whether a computer program is patentable until 1972. In that case, the Court reviewed a patent application describing an invention as a method for converting binary-coded decimal numerals into pure binary numerals in a general-purpose digital computer. Although the Court found that the invention was not patentable, it stated that this decision did not preclude ever finding a computer program patentable.

In the 1972 United States Supreme Court case, Gottschalk v. Benson, the Court framed the issue as: “The question is whether the method described and claimed is a ‘process’ within the meaning of the Patent Act.” The Court went on to describe the claimed invention:

The patent sought is on a method of programming a general-purpose digital computer to convert signals from binary-coded decimal form into pure binary form. A procedure for solving a given type of mathematical problem is known as an ‘algorithm.’ The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another. From the generic formulation, programs may be developed as specific applications.

Prior United States Supreme Court cases had interpreted the word “process” in Section 101 of the Patent Act to preclude patents on “scientific truth or the mathematical expression of it.” Accordingly, the Court in Benson decided the claimed invention was just a mathematical algorithm, and thus not within the subject matter of patentable inventions as a “process.”

The wake of the Benson decision caused all sorts of difficulties with the word “algorithm.” These difficulties arose out of decisions of Patent Examiners who rejected patent applications because an application contained the word “algorithm” or the examiner used an overly expansive definition of “mathematical algorithm.”

In 1976, the Court of Customs and Patent Appeals (the “CCPA”) reviewed a patent application that claimed an invention for dynamically rearranging priorities in a multi-program operating system. The PTO Board of Appeals had taken the position that Benson precluded patentability of computer programs. The CCPA decided differently, limiting the Benson decision to precluding patentability of mathematical algorithms, but not computer programs generally. The CCPA was careful to explain the difference under Patent Law (as it understood it in 1976) between a mathematical algorithm and general algorithms:

Over-concentration on the word ‘algorithm’ alone, for example, may mislead. The Supreme Court carefully supplied a definition of the particular algorithm
The broader definition of algorithm is ‘a step-by-step procedure for solving a problem or accomplishing some end.’ Webster’s New Collegiate Dictionary (1976). It is axiomatic that inventive minds seek and develop solutions to problems and step-by-step solutions often attain the status of patentable invention. It would be unnecessarily detrimental to our patent system to deny inventors patent protection on the sole ground that their contribution could be broadly termed an ‘algorithm.’

In 1978, the CCPA reviewed a patent application that claimed a new method for typesetting alphanumeric information (such as a mathematical expression) using a computer-based control system (employing a hierarchical tree data structure) in conjunction with a phototypesetter. The PTO had rejected the application on the basis that Benson precluded patentability of an invention where the only “novel” part was the computer program. Once again the CCPA stated that Benson only precluded patentability of mathematical algorithms, but in this case, the claims were broader algorithms:

The ‘local positioning algorithm’ described in appellant’s specification is the order of steps in processing the hierarchical tree structure and spatially relating the various characters to be displayed. Appellant has thus used the term ‘algorithm’ as a term of art in its broad sense, i.e., to identify a step-by-step procedure for accomplishing a given result.

Only a few months later, the CCPA again had to reverse a decision of the PTO Board of Appeals that rejected a patent application for a computer program. The PTO Board of Appeals had used the broad definition of “algorithm” instead of the narrower definition of “mathematical algorithm” in its interpretation of Benson.

The invention in this case claimed a method of translating from one natural language (e.g. Russian) to another natural language (e.g. English). The CCPA held that the process described did not solve a mathematical problem, directly or indirectly, and thus was not precluded from patentability.

1981 - The Supreme Court speaks again: Diamond v. Diehr

Even by 1980, the PTO was still trying to argue that an invention that claimed a stored program in a computer was not patentable. The United States Supreme Court, in a case called Diamond v. Diehr, reviewed a patent application for molding raw, uncured synthetic rubber into cured precision products including constant measurement of actual temperature inside the mold during the curing process to recalculate the cure time (by a computer program). The PTO examiner had rejected the claims (in 1976) on the basis that Benson precluded patentability of the steps of the process that used a computer program, and the remainder of the steps were already known in the industry. The appellate Court (CCPA) reversed the finding, but the PTO Commissioner requested the United States Supreme Court to review the decision in 1980. The United States Supreme Court addressed only the issue of whether a process employing a mathematical algorithm is patentable. The Court found that the invention used a mathematical
algorithm, but as a whole, the invention directed itself to the process of molding rubber and was thus patentable.

Because the invention in the patent application in *Diehr* did not encompass a general non-mathematical algorithm (or so says the Court), the Court stated that this decision was confined to mathematical algorithms:

The term ‘algorithm’ is subject to a variety of definitions. The petitioner defines the term to mean:

1. A fixed step-by-step procedure for accomplishing a given result; usually a simplified procedure for solving a complex problem, also a full statement of a finite number of steps.
2. A defined process or set of rules that leads [sic] and assures development of a desired output from a given input. A sequence of formulas and/or algebraic/logical steps to calculate or determine a given task; processing rule. [citation omitted].

This definition is significantly broader than the definition this Court employed in *Benson* and *Flook*. Our previous decisions regarding the patentability of ‘algorithms’ are necessarily limited to the more narrow definition employed by the Court, and we do not pass judgment on whether processes falling outside the definition previously used by this Court, but within the definition offered by the petitioner, would be patentable subject matter. 35

*A history of the lower courts trying to understand*

So where did the Supreme Court decisions leave us regarding the patentability of software-related inventions? In August 1982, the CCPA reviewed a patent application for a method of controlling the internal operations of a programmed computer that allow for the execution of several formulas independent of the order in which the steps are presented. 36 This particular patent application apparently had a long and tortured treatment: it was originally filed with the PTO in 1970 (two years before the *Benson* decision). The applicants had used the word “algorithm” to describe the invention. The PTO examiner rejected the application in 1972 on the basis that *Benson* precluded “algorithms.” The PTO Board of Appeals affirmed the rejection on the basis that *Benson* precluded “algorithms.” The PTO Board of Appeals affirmed the rejection on the basis that the invention was obvious. The CCPA reversed, finding that the invention claimed no mathematical algorithm (even if it manipulated mathematical algorithms) and that *Benson* and *Diehr* did not preclude patentability of computer programs generally. 37

---

In September 1982, the CCPA reviewed a patent application for a process and apparatus for testing a complex system and analyzing the results according to a factor of function or malfunction (illustrated as a process for diagnosing neurological disorders in patients). The CCPA held the claims were not patentable because they really distilled down to a mathematical algorithm representing a mental process that is not applied to physical elements or process steps.

In 1989 the Federal Circuit Court of Appeals (the successor court to the CCPA) again held that claims for a method of testing a complex system to determine whether the system condition is normal or abnormal and to determine the cause of any abnormality was non-patentable as a mathematical algorithm. The only “physical” step in the invention was the gathering of data for input, a step which was not significant enough to take the invention out of the category of unpatentable mathematical algorithms. Three days later, the Federal Circuit Court of Appeals held that an auto-correlation circuit for use in voice pattern recognition was patentable despite the use of computer algorithms. In this second case, the Court emphasized that the overall invention was the apparatus (the interrelation of the various hardware components which operate according to an algorithm) and not the algorithm itself.

The next case from the Federal Circuit was issued in March of 1992, Arrhythmia Research Technology, Inc. v. Corazonix Corporation. That case originated in the District court after the patent had already issued. The invention was for a method of analyzing electrocardiograph signals using a computer to diagnose heart attack patients who are at high risk for heart arrhythmia. The lower court had held the patent was invalid because the mathematical algorithm used to detect certain heart signals predominated. The Federal Circuit disagreed, holding that the overall result (not the process) controlled: in this case, the conversion of one physical electrical signal to another signal.

This distinction between the method and its result appears to indicate a more liberal view from the Federal Circuit on the issue of patentability of computer programs. However, the PTO continues to err on the side of conservatism. Presently, one of the policies of the PTO has been articulated as:

Claims that deal with ‘computer programs,’ equations or mathematical calculations, or methods which might be suspect under the Supreme Court’s exclusions . . . need to be carefully analyzed. . . . If the claim is the least bit suspect regarding the 101 question—make the rejection. . . . The whole ‘software’ question is very sensitive. . . . If too many patents are issued that are borderline or that are perceived as being borderline, the unfriendly publicity could adversely affect us.

Recent Federal Circuit Decisions

More recently, the Federal Circuit has issued several cases dealing with the question of whether various software-related inventions fall outside the areas of
proper subject matter as unpatentable mathematical algorithms. In the decision of In re Schrader, the court found a competitive bidding method to include a mathematical algorithm, which rendered the claims unpatentable. The court found that even though no mathematical formula was explicitly stated in the claims, the method implied the use of a simple summing method without physical elements or process steps.

However, after a very divided court issued the case of In re Alappat, the future appeared brighter for the viability of broad computer software patents. The claims of that upheld patent were directed toward a “machine” for smoothing waveforms for an oscilloscope. Since the claims were directed toward a machine, including elements described in a “means-plus-function” format, it appeared that claim drafting techniques would be the answer to making software-related inventions patentable subject matter. The case pointed out that although specific hardware structures had been disclosed in the specification, (ALU, ROM, shift registers) the claim at issue would also read on a general-purpose computer programmed to carry out the invention.

The case of In re Warmerdam seemed to support this view since the court found the robotic collision avoidance system “method” and “data structure” claims of that patent to be unpatentable subject matter, yet upheld the claim directed toward a machine including memory generated according to the unpatentable method. Another interesting note regarding that case is that the court avoided the issue of determining a definition for a “mathematical algorithm,” instead finding the issue to be whether the method for generating a data structure representing the shape of a physical object was “a process that goes beyond simply manipulating ‘abstract ideas’ or ‘natural phenomena’.”

Later in the year, the Federal Circuit decided the case of In re Lowry to reverse a decision by the Board of Patent Appeals and Interferences to find a claim to a “memory” with a particular “data structure” for organizing memory to be patentable subject matter. However, rather than presenting a direct argument under 35 U.S.C. § 101, the Board argued that the data structure was analogous to printed matter, and thus obvious under 35 U.S.C. §§ 102 and 103. The court disagreed and found the “memory” claims to be patentable.

Finally, the most recent and troubling case addressing the patentability of software-related inventions is the case of In re Trovato. In this case, the court found both method and machine claims directed toward planning a path for an object to follow in a physical task space to be unpatentable subject matter. The court upheld application of the Freeman-Walter-Abele test and found that although the claims did not express the mathematical algorithm in terms of a mathematical formula, the court re-stated that words in a claim operating on data can serve the same purpose as a formula.
Of particular importance to the court was the lack of disclosed structure in the specification, prompting the court to find the machine claims to be drafted in “illusory apparatus format.”52 Finally, the court noted that “Trovato does not claim to have invented a new kind of computer which the recited mathematical algorithm controls.53 Nor do they claim that the recited mathematical algorithm has been combined with a new memory controlling a computer known to the art.”54

In conclusion, In re Trovato seems to erase the perception created by the court in In re Warmerdam that claim drafting techniques will provide the entire answer for the persistent subject matter problems for software-related inventions. Instead, future specifications will apparently be scrutinized much more closely for detailed structural disclosure, and mathematical algorithms will apparently be found to exist in more unlikely locations.

The Patentability Spectrum

The case law to date indicates that the patentability of software-related inventions can be described by examining the spectrum of those inventions; the ends of the spectrum are clear, but the lines in the middle are very fuzzy:

Mathematical algorithms found not to be patentable by themselves.

- a computer software algorithm for converting from binary-coded decimal numerals to pure binary numerals (Benson)
- formula for computing an updated alarm limit during a catalytic conversion process (Flook)
- an improved mathematical equation for interpreting seismic waves (In re Walter)
- a competitive bidding method where simply summing was found to be a mathematical algorithm (In re Schrader)
- a method and a data structure for collision avoidance (In re Warmerdam)
- a method and a machine for planning object movement (In re Trovato)

Algorithms that are expressed as mathematical formulas found not to be patentable by themselves, and those algorithms which appear to replicate human decision-making skills found not to be patentable:

- a process and apparatus for testing a complex system and analyzing the results (for use in medical diagnosis), using functions of K elements (any arbitrary subdivision of the system such as neuropathways in the human nervous system) (In re Meyer)
- a process for determining “abnormalities” in a complex system (In re Grams)
Computer algorithms unrelated to mathematics found to be patentable:

- computer program for the translation of natural languages (*In re Toma*)
- computer program for executing several equations regardless of the order of their input (*In re Pardo*)

Computer algorithms that pertain to the operations of the hardware found to be patentable:

- system software (*Chatfield*)
- firmware (*Bradley*)
- algorithm for typesetting alphanumeric equations (*Freeman*)
- waveform smoothing algorithm for oscilloscope (*In re Alappat*)
- a machine with a collision avoidance memory (*In re Warmerdam*)
- a memory with data structure for organizing data (*In re Lowry*)

Processes or apparatus that use computer programs as a component of the overall invention found to be patentable:

- rubber curing process (*Diehr*)
- improved CAT scan procedure (*Abele*)
- apparatus for voice pattern recognition (*Iwahashi*)
- method for analyzing electrocardiograph signals (*Arrhythmia*)

Despite the various analyses by the Courts that result in this categorization of the cases, it is also clear that some of these cases have differences without real distinctions. The *Arrhythmia* case (1992) appears to be not different in real principle from the *Walter* case (1980) (the process of analyzing electrocardiograph signals is patentable although the process of analyzing seismic wave signals is not patentable). Further, it is difficult to really say that the “mathematical algorithms” in the two systems analysis cases reflect a natural truth (like the “law of gravity”), as the patent application claims read more on a methodology of estimating probabilities. Accordingly, it appears the analysis of “algorithms” and “data structures” needs further refinement as the Courts better understand the function and result of computer programs.

*Recommendations to the Computer Software Developer*

Regardless of your opinion on the issue of whether software-related patents are good or bad for society, the fact remains that a rapidly increasing number of software companies are filing software-related patent applications every day. Furthermore, the number of software-related patents issued by the PTO continues to increase. These facts create a need for software companies to begin developing programs and procedures to also protect their intellectual property through the patent system.
One of the reasons for building a patent portfolio is the common practice of patent cross-licensing. Among companies with patent portfolios, it is a very common practice for one company to offer to cross-license one or more patents of its own when accused of infringing a patent belonging to another company. Through this practice, the patent assets of both companies are increased. Another related reason for utilizing the patent system is to create a more effective defense to patent infringement suits. In other words, if another company is aware of your patents and fears retaliatory patent infringement actions, this knowledge can often provide a fairly effective "defense."

Thus, software companies should begin evaluating the best internal procedures for identifying potentially patentable ideas and pursuing patent protection for such ideas. Since software-related patent applications are often relatively expensive and time-consuming, appropriate business decisions should also be made to support such an endeavor.


---

7. 35 U.S.C. § 154. The law regarding the 17-year-from-grant term for utility patents will soon change to provide for 20-year-from-filing terms for utility patents. In other words, all utility patents issuing on applications filed after June 8, 1995, will have terms of twenty (20) years from the earliest effective filing date for the invention.
8. After June 8, 1995, there will also be a new type of patent application available: the provisional patent application. This application will be good for only one year and will not be examined, but the application will establish a prior date for filing a subsequent continuation without starting the twenty-year term, as well as provide the inventor an opportunity to use the label “patent pending.”
9. According to the new laws, the types of infringing activities will also specifically include “offering to sell” and “importing.”
However, the second requirement, that of whether the invention is the proper subject matter for consideration of patentability, is currently the primary issue of concern for software-related inventions.

As compared to all other devices within the field of “prior art” to the invention. 

In some cases, the first step of the process involves performing a patentability search in an effort to gauge the likelihood of actually receiving an issued patent for a particular invention.

International patent law differs significantly however in two important areas: 1) the first to file wins; and 2) there can be no public disclosure of the invention prior to filing the application.


A sequence of formulas and/or algebraic/logical steps to calculate or determine a given task; processing rules.” In re Toma, 575 F.2d 872 at note 4 (C.C.P.A., 1978)


In re Pardo, 684 F.2d 912 (C.C.P.A., 1982).

In re Meyer, 688 F. 2d 789 (C.C.P.A., 1982).

In re Grams, 888 F. 2d 835 (Fed. Cir., 1989).

In re Iwahashi, 888 F. 2d 1370 (Fed. Cir., 1989).


44 In re Schrader, 22 F.3d 290, 30 U.S.P.Q.2d 1455 (Fed. Cir. 1994).
46 31 U.S.P.Q.2d at 1558. “Consequently, a computer operating pursuant to software may represent patentable subject matter, provided, of course, that the claimed subject matter meets all of the other requirements of Title 35. In any case, a computer, like a rasterizer, is apparatus not mathematics.”
48 31 U.S.P.Q.2d at 1759.
49 In re Lowry, 32 U.S.P.Q.2d 1031 (Fed. Cir. 1994).
51 “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.” Arrhythmia, 958 F.2d at 1058, 22 U.S.P.Q.2d at 1037.
52 33 U.S.P.Q.2d at 1199.
53 This is an interesting comment in light of the court’s previous statement in In re Alappat 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.” Id. at 1558.
54 This final comment is apparently a reference to the court’s earlier decision in In re Lowry.