

Intellectual Property Protection of Software in the 21st Century

Mark E. Nikolsky, Esq.
Registered Patent Attorney
Wolff & Samson, P.A.
5 Becker Farm Road
Roseland, New Jersey 07068
Tel: 973-533-6504
Fax: 973-436-4407
e-mail: mnikolsky@wolffsamson.com

I. INTRODUCTION

The software industry has, without doubt, undergone a tremendous period of growth in the last decade. Traditionally, copyright and trade secret law comprised the main forms of intellectual property (“IP”) protection available for computer software. Recently, however, the number of patent applications filed for software-related inventions has increased, and the United States Patent and Trademark Office (“USPTO”) continues to seek examiners with software-related backgrounds to handle the increased caseload. Further, the increased popularity of the Internet, in both consumer and corporate contexts, has led to an increase in business method patents relating to Internet-based technologies. The purpose of this article, therefore, is to provide the software development community with an understanding of IP law and software patents in particular, and to offer some preliminary guidance as to how software developers may protect their creations.

II. IP FORMS OF PROTECTION FOR SOFTWARE

Any software developer, whether an individual or an organization, should be aware of the various forms of IP protection that are available for software-related innovations. In general, IP law can be thought of as comprising three main areas: copyright, trademark, and

patent law. Each of these areas protect various aspects of intellectual property, and each can serve a distinct role in protecting software innovations.

A. Copyright

Copyright law protects an author's work from unauthorized copying, and grants to the author the exclusive right to reproduce, prepare derivative works from, distribute copies of, and perform his or her work.¹ These rights apply to, among other things, literary, musical, dramatic, pictorial, graphic, audiovisual, sound, and architectural works, in addition to software. In order to procure such rights, an author must reduce his or her work to a tangible medium, such as on paper, CD-ROM, disk, tape, computer file, or other medium, and include a copyright indication (usually the © symbol), followed by the author's name and the date of creation of the work. Thus, basic copyright protections are available at the time of creation of the work and reduction of same to a tangible medium.

However, in order to enjoy national rights and to prevent others from copying or reproducing one's work anywhere within the United States, it is necessary to procure a federal copyright. This can be achieved by filling out an appropriate copyright application with the U.S. copyright office, in addition to a portion of the work to be protected (depending upon the type of work) and the associated government filing fee. The registration forms to be used will depend upon the type of work to be registered, but in general, computer programs can be registered using the Form TX application. These forms are available on the U.S. Copyright Office's website at www.loc.gov/copyright.

¹ See generally 17 U.S.C. § 106, et seq. (West 2002).

B. Trademark

Trademarks protect marks that identify goods or services used in commerce.² Examples of trademarks include ORACLE[®] for database and software applications, owned by Oracle, Inc., and EBAY[®] for Internet auction services, owned by Ebay, Inc. In order to procure national trademark rights for goods or services, it is necessary to prepare and prosecute a trademark application with the USPTO. Not all marks, however, can be registered. For example, any mark that is generic (*i.e.*, “Milk” for milk), can never be registered. Further, any mark that is confusingly similar with any other existing mark may not be registered. Thus, a USPTO examiner will often reject a trademark application based on a likelihood of confusion between the mark sought to be protected and one or more existing marks, and the services of a skilled trademark attorney will be necessary to overcome such rejection.

Once a trademark is allowed, the holder may prevent another person or entity from using any mark that is confusingly similar to the registered mark. In the software context, trademark protection may be available not only for a particular software product, but also the entity responsible for creating the product. Further, service mark protection (service marks identify services, and are akin in nature to trademarks, which protect the identity of goods) may be available for software consulting services.

C. Patent

Patents, by far, represent the most complex part of IP law, but potentially afford some of the most valuable protections for software. Patents protect new and useful inventions, and

² See generally 15 U.S.C. § 1051, et seq. (West 2002).

grant to an inventor the exclusive right to make, use, and sell an invention anywhere within the United States. Patent protection may be available for any new and useful process, machine, manufacture, composition of matter, or any new and useful improvement thereof.³ A patent may not be granted if the invention has been publicly used or offered for sale in the U.S. more than one year before the filing of an application for patent. Further, if patent protection is desired in foreign countries, the invention must not be publicly disclosed whatsoever prior to filing in some foreign countries.

In order to procure patent protection for a given innovation, it is generally recommended that a patent first search be conducted and a patentability opinion be prepared by legal counsel. If it is determined that the invention is novel and non-obvious in view of the prior art, the inventor may then proceed with preparing and filing a patent application with the USPTO or in a foreign country. Once the application has been prepared and filed, the USPTO will quite often reject the application based upon prior patents and references that, either alone or in combination, either anticipate or render the invention obvious in view of the prior art. A patent attorney may then prepare one or more responses to the USPTO examiner's arguments, in order to seek allowance of the patent.

When a patent issues, the term of protection extends for 20 years from the date of filing for the patent. Software-related inventions, including programs, algorithms, processes, and business methods, may be capable of patent protection, and may provide a software developer with an effective tool for protecting his or her creation. Although a patent can never protect a fundamental scientific truth, a law of nature, or a bare idea having no real-world application, if

³ See generally 35 U.S.C. § 1, et seq. (West 2002).

a particular software innovation has some concrete, tangible, and useful result, it may indeed be protectable by a patent.

III. PATENT PROTECTION OF SOFTWARE -- A BRIEF HISTORY

While it is generally accepted today that software can, and indeed often is, be protected by patent law, it is interesting to trace the history of software patents. The present attitude of the Courts toward software patents can be traced to early patents containing mathematical algorithms as elements of the claimed invention. A distinct paradigm shift with regard to the Courts' attitudes can be discerned, beginning with the case of MacKay Radio & Telegraph Co. v. Radio Corp. of America.⁴ In this case, perhaps the first to address the patentability of a mathematical algorithm, the claimed invention related to a novel radio antenna.⁵ The claim language of the patent, while directed to claiming the antenna itself, also expressed a mathematical formula for determining a desired angle and length of antenna wire elements.⁶ Specifically, the applicant claimed a "formula in empirical form for determining the desired angle [of the antenna] when wavelength and length of wire are known."⁷ The alleged infringer challenged the validity of the patent, asserting that the formula represented an unpatentable scientific truth. The Court, careful to distinguish between unpatentable scientific truths from patentable subject matter, noted that "[w]hile a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid of

⁴ See MacKay Radio & Tel. Co. v. Radio Corp. Of America, 306 U.S. 86 (1939).

⁵ See id.

⁶ See id.

⁷ Id. at 95.

knowledge of scientific truth may be.”⁸ Thus, the Court concluded that although any claims directed to the formula for calculating the angle of the antenna were, alone, unpatentable, the antenna design resulting from the calculation constituted patentable subject matter. Notwithstanding, the patent was not held infringed.

It was not until 1972 that the Supreme Court first addressed the patentability of computer programs. In Gottschalk v. Benson, an invention directed to a process for converting a binary-coded decimal (BCD) number to a binary number was challenged on the grounds of statutory subject matter.⁹ In contrast to the patent at issue in MacKay, the claims of which were directed to an apparatus, the claims of Gottschalk were directed solely to a process; namely, the conversion of BCD values to binary values.¹⁰ Regardless of this difference, the Court concluded that the claimed process was so “abstract and sweeping as to cover both known and unknown uses of the BCD to pure binary conversion[,]” and that it could be “performed through any existing machinery or future-devised machinery or without any apparatus.”¹¹ Accordingly, the court concluded that because the invention, as claimed, was so broad in scope, it did not define a patentable process within the meaning of the patent statute.¹² Further, the Court concluded that the claimed algorithm was not patentable, and suggested that

⁸ Id. at 94.

⁹ See Gottschalk v. Benson, 409 U.S. 63 (1972).

¹⁰ See MacKay, 306 U.S. at 86 (1939).

¹¹ See Gottschalk, 409 U.S. at 68 (1972).

¹² See generally 35 U.S.C. § 101 (West 2001).

the “patent laws should not be extended to cover these programs.”¹³ Wary of such an overarching position, however, the Court noted that:

We do not hold that no process patent could ever qualify if it did not meet the requirements of our prior precedents. It is said that the decision precludes a patent for any program servicing a computer. We do not so hold.¹⁴

Thus, Gottschalk, in conjunction with MacKay, appears to cast a palpable aversion to patenting computer programs, especially considering the fact that mathematical algorithms, defined by the Court as unpatentable subject matter, are closely inter-related with computer software.

This sentiment continued throughout the 1970's, culminating with the case of Parker v. Flook.¹⁵ In Parker, the claimed invention related to a process for updating alarm limits for a catalytic chemical conversion process. Specifically, the first claim of the patent recited a “method for updating the value of at least one alarm limit” using a mathematical algorithm.¹⁶ The claimed process could be carried out by simple hand calculation, or by utilizing a computer. The Court concluded that the recited invention did not fall within the scope of patentable subject matter, and that the patent was invalid. Specifically, the Court noted that “Respondent's application simply provide[d] a new and presumably better method for calculating alarm limit values,” thereby rendering the invention akin to an unpatentable law of

¹³ See Gottschalk, 409 U.S. at 72 (citing To Promote the Progress of ... Useful Arts, Report on the President’s Commission on the Patent System (1966)).

¹⁴ Id. at 71.

¹⁵ See Parker v. Flook, 437 U.S. 584 (1978).

¹⁶ Id. at 596-97.

nature.¹⁷ Further, it appears that the Court distinctly ruled out improvements to mathematical algorithms as statutory subject matter.

In 1981, a distinct paradigm shift appeared in Supreme Court jurisprudence relating to software patents. In Diamond v. Diehr, the claimed invention related to a process for curing synthetic rubber that included an algorithm for determining curing time.¹⁸ The Court placed particular importance on the fact that the claimed invention had other, non-mathematical elements, and was generally directed to a rubber curing process, as opposed to the specific algorithm forming a part thereof.¹⁹ In particular, the Court noted that “a claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program, or digital computer.”²⁰ Further, the Court noted that “[t]ransformation and reduction of an article ‘to a different state or thing’ is the clue to the patentability of a process claim that does not preclude particular machines.”²¹ Thus, the Court, for the first time, appeared to shift the focus of its attention to whether the claimed invention, as a whole, defines a patentable invention, notwithstanding the presence of algorithms in the claims of the application.²² Additionally, this case also raises the prospect that a software-based invention may qualify as statutory subject matter if it is directed to some type of concrete, useful, and tangible result, even if the invention comprises an algorithm as a central

¹⁷ Id. at 594-95.

¹⁸ See Diamond v. Diehr, 450 U.S. 175 (1981).

¹⁹ See id.

²⁰ Id. at 187.

²¹ Id. at 184 (citing Gottschalk v. Benson, 409 U.S. 63 (1972)).

²² See generally id.

component. Rather clearly, such a prospect dovetails comfortably with the Supreme Court’s expansive construction of patentable subject matter as including “anything under the sun that is made by man....”²³

The Courts appeared to be silent as to the fate of software patents during the remainder of the 1980’s, for no significant cases appear in this timeframe. In 1994, however, the Court of Appeals for the Federal Circuit (“CAFC”) decided numerous cases that expanded the reach of patentable subject matter in the software realm. For example, in In re Allapat, the CAFC held that a rasterizing algorithm for digital oscilloscopes, claimed in means-plus-function language, defined patentable subject matter, and provided that the critical inquiry for software cases is whether “the claimed subject matter *as a whole* is a disembodied mathematical concept....”²⁴ The CAFC further held, presumably to the delight of the software community, that computer “programming creates a new machine, because 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.”²⁵

The apparent pro-software paradigm shift initiated by the Court in In re Allapat continued in additional software-related cases decided by the CAFC. In In re Lowry, the CAFC concluded, in contrast to a prior decision to the contrary,²⁶ that a computer data structure

²³ See Diamond v. Chakrabarty, 447 U.S. 303 (1980); see also Hearings on H.R. 3760 before Subcomm. No. 3. Of the House Comm. On the Judiciary, 82d Cong., 1st Sess. 37 (1951).

²⁴ In re Allapat, 33 F.3d 1526, 1544 (Fed. Cir. 1994) (emphasis in original).

²⁵ Id. at 1545.

²⁶ See generally In re Warmerdam, 33 F.3d 1354 (Fed. Cir. 1994) (holding that a method for providing a computer data structure did not constitute patentable subject matter; rather, the claimed invention represented an unpatentable, paradigmatic abstract idea that resulted in basic mathematical constructs.)

expressed as an apparatus claim could constitute patentable subject matter.²⁷ The invention at issue related to an object-oriented data structure capable of storing hierarchical and non-hierarchical relationship information.²⁸ The Court held that because the claimed invention was directed to an apparatus (namely, a data structure embodied as memory within a digital computer system), the invention fell within the bounds of patentable subject matter.²⁹ Shortly thereafter, however, the CAFC limited its seemingly unqualified approval of data structures by holding that such structures do not automatically define statutory subject matter when the applicant constructs claims having unsubstantiated reference to supporting computer hardware.³⁰

The above cases demonstrate a gradual acceptance of software patents by the Courts. Perhaps in response to the utter explosion of the Internet and Internet-based software inventions, the CAFC, in 1998, finally declared that business methods embodied in software could qualify as patentable subject matter.³¹ In response to this decision, a wide array of business method patents began to appear, many of which related to Internet-enabled software systems. Thus, it appears that software patents had gained approval by the late 1990's, and that a paradigm shift toward gradual acceptance of software patents occurred.

²⁷ See In re Lowry, 32 F.3d 1579 (Fed. Cir. 1994).

²⁸ See id.

²⁹ See id.

³⁰ See In re Trovato, 42 F.3d 1376 (Fed. Cir. 1994).

³¹ See State Street Bank & Trust Co. v. Signature Financial Group, 149 F.3d 1368 (Fed. Cir. 1998).

IV. CONCLUSION:
POTENTIAL QUESTIONS FOR SOFTWARE DEVELOPERS

Software developers may consider the following questions helpful in addressing potential IP issues with relation to software:

(a) Does the software application employ some new and useful process, device, or other innovation that does not exist in the art? If so, consider seeking patent protection on the process, device, or innovation. Thoroughly document each step of the invention, preferably in a lab notebook or some other means of recording the project. Refrain from publicly using or offering for sale the invention until the decision whether to seek patent protection has been addressed. Perform preliminary searches in industry texts, databases, and USPTO records to determine whether the innovation contributes something new and useful over the prior art. Prepare flowcharts, state diagrams, and other documentation relating to the core features of the innovation. Consult patent counsel to do a patent search and/or prepare and prosecute an application for patent.

(b) Has the innovation been reduced to a tangible medium of expression, *i.e.*, code, CD-ROM, or text? If so, consider filing federal copyright applications on each embodiment of the code, including successive revisions. Include copyright notices, dates of creation, and author names in the code.

(c) Has the innovation been created using an employer's resources, or on company time? If so, respective IP rights of the employer and the developer should be carefully ascertained by an attorney prior to seeking IP protection on the innovation.

(d) Will a saleable product and/or service result from the innovation? If so, consider seeking one or more trademark registrations for the product and/or service. Prior to

selling the product or engaging in contractual relations with other individuals or entities, carefully review the potential rights and liabilities that may be present with an attorney. Consider having an attorney draft pertinent contracts for sale, licensing, or assignment of the innovation.

In view of the above discussion of the potential IP protections available for software, developers should consider all areas of IP law that may be applicable to protect software programs, applications, and services. Early thought and careful planning of an IP strategy for any project may, in the future, be of great value to the software developer.