

Software Does Not Exist Without a Computer:
A Case for Patentability in China and the United States

By Eric L. Sutton¹

Both the U.S. and China require a practitioner to claim supporting hardware in a software-related patent. By selling only the software, infringers can easily design around software-related patents. Patent law is designed to promote the progress of the useful arts, including computer science, finance, and management. Therefore, patent law should appropriately reward software innovators for their contributions to the field by allowing them to claim software without hardware.

Generally, a function is a set of instructions that work together to affect the computer display, settings, and memory. Functions are normally configured to work on a fairly specific computer arrangement, such as at a certain resolution, speed, bandwidth, etc. Software is a group of functions that work together, usually dependant on an input such as the mouse, keyboard, or any device adapted to be manipulated by the environment. A computer program is equivalent to software. From a more complex view, a group of instructions is a function; a group of functions is a class; a group of classes is a library; and, a computer program is the main programming loop that acts on the library. Software, under the more complex view, could be defined as a group of computer programs that allow a higher task to be performed on a variety of platforms. For the sake of consistency and simplicity, though, software or a computer program will be referred to herein as a collection of functions.

¹ 3L student, The John Marshall Law School.

China allows inventors to protect software under certain conditions. First, the software must be claimed along with supporting hardware. Second, the software must use technical means to solve technical problems and result in technical effects. China does not recognize fields like “management” and “finance” as technical fields, but systems, machinery, and electronics are considered technical fields. Thus, software-related business methods are generally not patentable in China unless the software provides a technical solution beyond the financial effect for a business. *Presentation by Dr. Lulin Gao for Patent Law and Practice in China* (September 2007).

U.S. patent law protects anything useful that is made by man. *Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1998). The U.S. has the broadest scope of patent protection in the world, but courts have not given the utility requirement its broadest possible interpretation. In 2007, the Federal Circuit narrowly held that "a claim reciting an algorithm or abstract idea can state statutory subject matter *only if, as employed in the process, it is embodied in, operates on, transforms, or otherwise involves another class of statutory subject matter*, i.e., a machine, manufacture, or composition of matter." *In re Comiskey*, 2007 U.S. App. LEXIS 22414, *27 (Fed. Cir. 2007) (emphasis added) (quoting *Parker v. Flook*, 437 U.S. 584, 588 (1978) for the proposition that a method containing an algorithm is patentable subject matter only if “the process 'either [1] was tied to a particular apparatus or [2] operated to change materials to a 'different state or thing'”; citing *Diamond v. Diehr*, 450 U.S. 175, 184 (1981) for the proposition that “a process claim reciting an algorithm could state statutory subject matter if it: (1) is tied to a machine or (2) creates or involves a composition of matter or manufacture”).

However, the Federal Circuit has previously left the door open for a broader scope of protection for computer software by noting, in *AT&T Corp. v. Excel Communs., Inc.*, that “the process of manipulation of numbers is a fundamental part of computer technology,” and “a mathematical algorithm may be an integral part of patentable subject matter such as a machine or process if the claimed invention as a whole is applied in a ‘useful’ manner.” 172 F.3d 1352, 1356-1357 (Fed. Cir. 1999) (citing *State St. Bank & Trust Co. v. Signature Fin. Group*, 149 F.3d 1368, 1373 (Fed. Cir. 1998)). According to *State St. Bank & Trust Co.*, a mathematical algorithm is part of an invention when it has a “practical application, i.e., ‘a useful, concrete and tangible result.’” 149 F.3d at 1373. *In re Alappat* held that “more than an abstract idea was claimed [in *Alappat*] because the claimed invention [in *Alappat*] was directed toward forming a specific machine that produced the useful, concrete, and tangible result of a smooth waveform display.” *AT&T Corp.*, 172 F.3d at 1357 (citing *In re Alappat*, 33 F.3d 1526, 1544 (Fed. Cir. 1994)).

In short, software will be rejected by the U.S. Patent and Trademark Office if it is characterized as a law of nature, a physical phenomenon, or an abstract idea, and software will be allowed only if it produces a useful, concrete, and tangible result. Software has been rejected in the U.S. as an abstract idea. See *Flook*, 437 U.S. at 588 (1978) (method for computing an “alarm limit” unpatentable); *Gottschalk v. Benson*, 409 U.S. 63, 71-72 (1972) (algorithm for converting binary code decimal numbers to equivalent pure binary numbers unpatentable). On the other hand, “a smooth waveform display” qualifies as a “useful, concrete, and tangible result” necessary for patentability. *In re Alappat*, 33 F.3d at 1537, 1544 (Fed. Cir. 1994).

In the U.S., a patent practitioner still needs to claim something concrete and tangible in a claim. Numbers, binary code, words, and written code are likely to be characterized as “abstract ideas” that are not “tangible.” See *Benson*, 409 U.S. at 71-72. Thus, software patent claims should include more than just how the code interacts with other code. Software patents need to claim code that acts on a device, such as a display, and not merely memory that can instead be represented as a set of intangible binary numbers. See *In re Alappat*, 33 F.3d at 1544. Thus, the U.S. practitioner still needs to include something technical in his claims, whether it is a simple device such as a display or a complicated device such as a processor.

Although the U.S. allows a broader scope of software-related patents than China, both the U.S. and China need to apply their current patentability standards more consistently so they no longer discriminate against the sciences of computers, finance, and economics. If an invention is referred to as “software” or a “computer program,” then it necessarily includes a computer. A computer normally includes at least a keyboard, a display, memory, and a processor. An algorithm can be performed by a human without a computer; software cannot be implemented without a computer.

Therefore, if a practitioner claims software that produces a field of numbers, then that claim implicitly requires code readable by a computer to accept input and process a field of numbers. If that practitioner further claims software that produces a field of numbers for analysis, then that claim implicitly requires that the invention display the field of numbers to a user or pass the field to another function for analysis. If the claim does not implicitly include a computer of some type, then “software” or a “computer program” is not the correct word to use. Instead, the word “algorithm” or “equation”

would be used. Neither the courts in China nor the courts in the U.S. have grappled with this language distinction.

The science of mathematics will not be hindered if a new and non-obvious mathematical algorithm can be written in a book but not performed on a computer to produce a practical and usable result. Patent law does not protect abstract ideas. In fact, it is impossible to stop others from thinking about a certain concept. *Rubber-Tip Pencil Co. v. Howard*, 20 Wall. 498, 507 (1874) (holding, "[a]n idea of itself is not patentable"); *Le Roy v. Tatham*, 14 How. 156, 175 (1853) (holding, "[a] principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right"); *Benson*, 409 U.S. at 67 ("[p]henomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and technological work"). Human beings have a right to think freely, but they do not have a right to freely give instructions to a computer so that the computer can perform functions. See *In re Alappat*, 33 F.3d at 1544 (holding, an invention "applied in any manner to physical elements or process steps" qualifies as producing a useful, concrete, and tangible result).

Although computer instructions should qualify as patentable subject matter, these instructions still need to satisfy the novelty and nonobviousness requirements. If a thought, equation, or algorithm is publicly known in the minds of scientists, then novelty and nonobviousness bars will prevent a patent on the application of the idea to computer software. The focus for the patentability of computer software should be on whether the software is new and nonobvious rather than whether it appropriately

functions with a machine. After all, the definition of “software” and “computer program” includes a machine. If the word “software” or “computer program” is used, then this should satisfy the “concrete” and “tangible” portions of the test for patentable subject matter. “While the mere use of the machine to collect data necessary for application of the mental process may not make the claim patentable subject matter, th[e] claims in combining the use of machines with a mental process, claim patentable subject matter.” *In re Comiskey*, 2007 U.S. App. LEXIS 22414 at *37 (quoting *In re Grams*, 888 F.2d 835, 839-840 (Fed. Cir. 1989)).

The word useful still has other limitations. The literal meaning of “useful” is “capable of being put to use,” or capable of serving a particular service or end. See Merriam-Webster Online Dictionary, *Definition of useful*, <http://www.m-w.com/dictionary/useful> (accessed Sept. 24, 2007). Thus, an invention that does not work is not useful. Also, an invention must have real-world utility at the time of filing and cannot just be interesting research material. *Brenner v. Manson*, 383 U.S. 519, 535. In *Diamond v. Chakrabarty*, 447 U.S. at 309, the U.S. Supreme Court said “anything under the sun that is made by man” is patentable. Patentability should not be limited beyond a function human beings have reason to perform, whether or not that function is being performed by the human on a computer. By not limiting what is patentable, the law promises an exclusive right to innovators of useful methods in all fields, thereby encouraging the progress of useful methods in society.

China has built more barriers to rewarding innovators by requiring that software use technical means to solve technical problems and result in technical effects. Software innovators in fields like “management” and “finance” are not rewarded for their

efforts to create methods that help businesses run more efficiently and profitably.

Businesses in China must rely on secrecy to protect any value in management and finance inventions that do not also touch on some technological field. Smaller businesses, including managers that relate most personally to the workings of the business, have no incentive to develop management and finance technology because they cannot protect innovations from competitors. Similarly, universities and academics have less of an incentive to enter a discourse on business efficiencies if there is no way for the institution to reap a benefit. Business and management innovations cannot be properly licensed and shared if the innovator is afraid that his idea will be stolen. If the innovator does not receive licensing fees, then he cannot fund his research. Currently, patent law in China tends to discourage independent research on business methods by failing to reward new, innovative methods with an exclusive right.

Business methods are generally patentable in the U.S., so long as they produce a “useful, concrete, and tangible” result. Business methods, like software, must act on something “concrete” and “tangible.” U.S. practitioners still need to claim a display, an input device, an output device, or some other hardware when patenting software or business methods that serve merely to inform the user. By requiring a practitioner to claim hardware, the law makes it easy for infringers to design around software patents and business methods. If the goal of the law is to encourage innovation in computer science, finance, and management, then the law should appropriately reward innovators for their contributions to the field. Currently, neither the U.S. nor China reliably rewards innovators for contributions to the fields of computer science, finance, and management.