



---

## As Software Patents Take Over, Expertise Is Key

Now that patents have eclipsed copyrights, counsel must have proper technical know-how.

BY JEFFREY R. KUESTER

SPECIAL TO THE NATIONAL LAW JOURNAL

The National Law Journal (p. B13)  
Monday, April 20, 1998

DISLIKE IT if you choose, but ignore it at your own risk. Patents, not copyrights, are now the only way to give adequate protection to the most important aspects of software. The "idea" behind a particular program is better protected as a patentable method of high-level steps rather than as a narrowly limited expression under a copyright scheme. Simply put, software copyright protection is nearly dead, but software patent protection is alive and growing.

Unfortunately, many attorneys and clients are reluctant to realize or adapt to this monumental shift in the intellectual property world. While there are many reasons for this reluctance, the resulting consequences for many will inevitably be disastrous. Software patents are now being asserted by more and more companies, and the numbers are staggering.

For example, IBM recently reported that nearly a third of its 1,724 patents issued in 1997 were software-related. The IBM patent portfolio generated about \$1 billion in annual royalties, up from \$350 million in 1993. Likewise, after losing the legendary \$120 million patent suit brought by the comparatively tiny Stac Electronics, Microsoft reacted with blazing speed by quickly building its software patent portfolio, receiving an amazing 199 patents in 1997 alone.

A typical scenario for many companies involves losing, or nearly losing, a major dispute before realizing the importance of patents. Since many disputes are resolved by cross-licensing patents owned by both parties, companies that have no real patent portfolios are severely disadvantaged in settlement talks.

Why are so many attorneys and clients reluctant to realize or properly react to this important IP development? One reason is the enormous difference in legal costs between preparing copyright applications and patent applications; spending more money on legal fees is always a difficult subject, even if the downside is potentially much more expensive.

Yet through decisions such as *Computer Associates v. Altai*<sup>1</sup> and *Lotus v. Borland*,<sup>2</sup> the courts have clearly demonstrated that copyright law now provides little protection for software. Also, since copyright law, unlike patent law, can vary among circuits, more uncertainty exists with respect to copyright law.

Conversely, in *In re Alappat*,<sup>3</sup> the U.S. Court of Appeals for the Federal Circuit made it clear that patentability is not lost on a patent claim that contains a mathematical formula or other nonstatutory matter if it also contains otherwise patentable subject matter. Following the court, the Patent and Trademark Office, or PTO, promulgated the Examination Guidelines for Computer-Related Inventions,<sup>4</sup> further opening the door to software

patentability. The number of software patents reportedly increased eightfold between 1977 and 1996.<sup>5</sup>

Though patents cost more, if a client is careful to choose a law firm that has a healthy number of experienced patent attorneys with electrical engineering and computer software degrees, the cost of preparing software patents can be controlled and accurately estimated.

### **Pursuing Patent Protection**

There are many considerations to keep in mind, most of which are distinctive to software patents. While it is now much easier to obtain such patents, the struggle to define statutory subject matter continues in many cases. Some approaches can lead to undesirable results.

Thus, the selection of a patent attorney for software patent applications is an important decision that should be based on the attorney's experience in drafting software patent applications and his or her understanding of software patents in particular. For example, a skilled attorney would already be implementing many important rules from the examination guidelines, one of which relates to an important restriction on the specification of the patent application.

That restriction is based on the new revelation that the determinative question in the statutory subject matter analysis, potentially taking place only when the patent is litigated, may well be whether the specification is directed to a specific machine or manufacture or, conversely, whether it is directed to any and every computer implementation of a process that has a practical application in the technological arts. The specification will be scrutinized for language corresponding to broadly expressed or to newly defined claim elements to determine the breadth to be assigned to those elements.

If the specification does not make it clear that the invention is directed to any and every machine for performing an underlying process, or to any and every manufacture--often claimed as a disk or other storage medium--that can cause a computer to perform an underlying process, then the specification could be interpreted as disclosing only a specific machine or manufacture.

While this would result in a relatively automatic finding of statutory subject matter, under Box 10 of the patent examiner's internal examination flow chart,<sup>6</sup> undesirable claim scope limitations may be the price paid. Conversely, broadening language in a specification may result in a finding of nonstatutory subject matter in the absence of a clearly disclosed and claimed practical application.

### **Affirmative Steps**

To address this dilemma, prudent software patent practitioners are drafting patent specifications that attempt both to avoid the undesirable claim scope limitations--based on a future finding of a specific machine or manufacture--and to disclose and claim unequivocally the practical applications of underlying processes. In addition, some practitioners are taking affirmative steps in the prosecution history of every software patent application to prevent the PTO from basing statutory subject matter findings on interpretations that claims are directed to specific machines or manufactures. Such a practice helps avoid the argument that an applicant essentially admitted the scope of a claim was limited accordingly.

Besides this important rule from the examination guidelines, there are many other observations that will be gleaned by observant software patent attorneys, including the following:

\* Claims should not be directed to descriptive material per se, whether the material is functional, such as a computer program, or nonfunctional, such as music; nor should claims be directed to nonfunctional descriptive material on a computer-readable medium, such as music on a disk.

\* Many claims to inventions that are not pure software can include elements introducing independent physical acts. Such acts include post-computer process activity or pre-computer process activity, i.e., the manipulation of data representing physical objects, or activities that achieve a practical application.

\* Claimed pre-computer process physical acts should be expressed as separate limitations in the claim to avoid being treated as ineffective on grounds that they are merely dictated by the performance of the mathematical operation itself. For example, "collecting" or "selecting" data for use in a claim may be treated as merely determining values for claimed variables, thus being ineffective to create statutory subject matter. Conversely, separate claim limitations of, for example, "creating" data for a process representing a practical application of a mathematical operation will be treated as limiting the claim beyond the mathematical operation. Such limitations may be treated as data gathering steps not dictated by the algorithm.

\* Claimed post-computer process physical acts should amount to a "significant use" of the solution of the claimed algorithm, rather than merely an output of the direct result of the mathematical operation, which would be ineffective in rendering the claim statutory.

\* If limiting a claim to a specific machine or manufacture is necessary, then including specific code segments (including pseudo-code) in the specification will help. Thus, in close cases, it would be wise to include such code segments in order still to be able to limit them to a specific machine or manufacture.

\* All claim elements that are defined, at least in part, by function should have a similarly named corresponding structure in the specification. Such a structure may be relied upon to show or disprove a practical application, so the specification should be drafted accordingly.

\* A claim should not merely convert one set of numbers into another set, but should be limited to a practical application. For example, a computer process which calculates a mathematical algorithm that happens to model noise may not be statutory, but a claimed process for digitally filtering noise employing the mathematical algorithm is statutory.

Software companies and general practice attorneys servicing them should start reacting to this pivotal change in IP law jurisprudence. Software companies should begin mounting aggressive programs to build their patent portfolios.

While an old adage in patent law is that ideas in the abstract are not patentable, it is now equally true that patents, rather than copyrights, provide software companies the closest thing to "getting the idea."

- (1) 982 F.2d 693 (2d Cir. 1992).
- (2) 49 F.3d 807 (1st Cir. 1995), *aff'd per curiam* by evenly split court, 116 S. Ct. 804 (1996).
- (3) 33 F.3d 1526 (Fed. Cir. 1994).
- (4) 61 Fed. Reg. 7478 (1996).
- (5) U.S. Patent and Trademark Office, Patent Counts by Class by Year (April 1997).
- (6) (<http://www1.uspto.gov/web/offices/pac/dapp/oppd/patoc.htm>). □

*Mr. Kuester is a partner at Atlanta's Thomas, Kayden, Horstemeyer & Risley L.L.P., a patent, copyright and trademark firm.*

-