

**This is an abridged version taken from  
*Legal Protection of Digital Information*  
Copyright © 2002, Lee A. Hollaar.  
A full, online version can be found at  
<http://digital-law-online.info>**

## **Chapter 5: Software-Based Inventions**

Probably no aspect of patent law has been more uncertain than the issue of what, if any, software-based inventions can be patented. But recent developments have gone a long way toward clarifying when a software-based invention will be statutory subject matter. The Patent Office has issued guidelines<sup>1</sup> for examiners that clarify when a software-based invention is statutory subject matter. Recent decisions by the Court of Appeals for the Federal Circuit make it clear that software-based inventions and methods of doing business are indeed patentable, although some uncertainty remains regarding the various forms for claiming such inventions.

The past lack of clarity came not from the statute itself, but from the court decisions that have attempted to interpret Section 101 of the patent laws,<sup>2</sup> which states what may be patented. The cases are difficult to reconcile and are often based on distinctions that don't stand up to technical (rather than legal) analysis. In many instances, an invention has been considered nonstatutory not because of the nature of the invention but because of the way it was claimed. In the prosecution of some patent applications for software-based inventions, most of the examiner's time has been spent on the question of whether the invention contained statutory subject matter, while for other patent applications, the question hasn't even been raised.

It is important to remember that whether a claimed invention recites statutory subject matter is just the first hurdle that must be conquered before one receives a patent. The invention must also be novel<sup>3</sup> and nonobvious,<sup>4</sup> and the application must fully disclose how to make and use the invention.<sup>5</sup> As an example, if an application today were to claim an ordinary pencil, it would clearly be claiming statutory subject matter, but no patent could issue because the claimed invention is not novel. On the other hand, if an application were to claim this text, it would be rejected because a writing is not statutory subject matter, regardless of its novelty or nonobviousness.

---

<sup>1</sup> 61 Fed. Reg. 7478 (1996).

<sup>2</sup> 35 U.S.C. §101.

<sup>3</sup> 35 U.S.C. §102.

<sup>4</sup> 35 U.S.C. §103.

<sup>5</sup> 35 U.S.C. §112, first paragraph.

## I. Reluctance at the Beginning

## II. Trying To Draw the Line

### II.A. After Diehr

### II.B. A New Clarity: The Alappat Decision

On July 29, 1994, the Court of Appeals for the Federal Circuit (the successor to the CCPA) decided en banc (all the judges of the court hearing the matter, rather than the more-common three-judge panel) on whether a rasterizer for graphical images was statutory subject matter. *In re Alappat*<sup>38</sup> was complicated by a question of the court's jurisdiction because the Commissioner of Patents had stacked the panel of the Board of Patent Appeals and Interferences after the original panel had produced a decision he thought went against the policies of the Patent Office. It also involved how means-plus-function claim elements should be interpreted during a patent examination.

After disposing of these two matters, the Federal Circuit noted that what was being claimed was a machine, albeit one that implemented an algorithm. There was no question that the machine was otherwise statutory subject matter.

However, because of the "mathematical algorithm" exception to statutory subject matter created by the Supreme Court, simply classifying the claimed invention as a machine did not end the analysis. A past court decision, *In re Johnson*,<sup>39</sup> stated that the exception "applies equally whether an invention is claimed as an apparatus or process, because the form of the claim is often an exercise in drafting."<sup>40</sup>

To determine the scope of the "mathematical algorithm" exception, Judge Rich examined the three Supreme Court cases on the patentability of software-related inventions:

A close analysis of *Diehr*, *Flook*, and *Benson* reveals that the Supreme Court never intended to create an overly broad, fourth category of subject matter excluded from Section 101. Rather, at the core of the Court's analysis in each of these cases lies an attempt by the Court to explain a rather straightforward concept, namely, that certain types of mathematical subject matter, standing alone, represent nothing more than *abstract ideas* until reduced to some type of practical application, and thus that subject matter is not, in and of itself, entitled to patent protection. . . .

Given the foregoing, the proper inquiry in dealing with the so called mathematical subject matter exception to Section 101 alleged herein is to see whether the claimed subject matter *as a whole* is a disembodied mathematical concept, whether categorized as a mathematical formula, mathematical equation, mathematical algorithm, or the like, which in essence represents nothing more than a "law of nature," "natural

---

<sup>38</sup> 33 F.3d 1526, 31 USPQ2d 1545 (1994).

<sup>39</sup> 589 F.2d 1070, 200 USPQ 199 (1978).

<sup>40</sup> 589 F.2d at 1077, 200 USPQ at 206.

phenomenon,” or “abstract idea.” If so, *Diehr* precludes the patenting of that subject matter.<sup>41</sup>

The court concluded that this was not the case in *Alappat*’s claimed invention, and so the “mathematical algorithm” exception for statutory subject matter did not apply.

The decision of the Board had also stated that one of *Alappat*’s claims was unpatentable merely because it “reads on a general purpose digital computer ‘means’ to perform the various steps under program control.” The Board decided that because the claim read on a programmed digital computer, it could be treated as a method claim rather than an apparatus claim.

The court found no basis for software-based inventions running on programmed general purpose computers to be per se unpatentable.

We have held that such 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.

Under the Board majority’s reasoning, a programmed general purpose computer could never be viewed as patentable subject matter under Section 101. This reasoning is without basis in the law. The Supreme Court has never held that a programmed computer may never be entitled to patent protection. Indeed, the *Benson* court specifically stated that its decision therein did not preclude “a patent for any program servicing a computer.” 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.<sup>42</sup>

*Alappat* allowed the Federal Circuit to restate and clarify its past decisions on whether software-related inventions are patentable. In particular, it is clear that a programmed general purpose computer must be regarded as a specialized piece of hardware both for determining whether a claim is drawn to statutory subject matter and when determining whether the invention is novel and nonobvious. It is also clear that the “mathematical algorithm” exception to statutory subject matter first discussed by the Supreme Court in *Benson* is limited to abstract mathematical concepts, not mathematics applied to a practical application. Machines, even though they carry out mathematical operations, are patentable.

This really did not differ substantially from the Patent Office’s practice. The time was long past when the Office rejected an application just because it was a software-related invention. There were over 10,000 patents that could be considered software-related at the time of *Alappat*. But the Office position had swung back and forth on the patentability of software-related inventions. *Alappat* restricts the Patent Office from treating software-related inventions more strictly under Section 101 than other inventions.

---

<sup>41</sup> 33 F.3d at 1543, 31 USPQ2d at 1556-1557.

<sup>42</sup> 33 F.3d at 1545, 31 USPQ2d at 1558 (citations omitted).

It is interesting to compare the Federal Circuit decision in *Alappat* with the CCPA decision in *Benson*, decided in 1971 and the first of the cases we've discussed here. Although the Supreme Court eventually held that *Benson* did not claim statutory subject matter, the CCPA thought differently.

Realistically, the process of claim 13 has no practical use other than the more effective operation and utilization of a *machine* known as a digital computer. It seems beyond question that the machines – the computers – are in the technological field, are a part of one of our best-known technologies, and are in the “useful arts” rather than the “liberal arts,” as are all other types of “business machines,” regardless of the uses to which their users may put them. How can it be said that a process having no practical value other than enhancing the internal operation of those machines is not likewise in the technological or useful arts?<sup>43</sup>

Both decisions were written by Judge Giles Rich, who, after serving on a federal court longer than any other judge, saw his position in *Benson* become the accepted law regarding the patentability of software-based inventions and other methods used with computers. Whether something has a particular use in the technological arts (has “utility”) is what now separates patentable methods from abstract and unpatentable mathematical algorithms.

## II.C. After Alappat

## II.D. The Patent Office's Guidelines

On June 2, 1995, following hearings on whether, and how, software-based inventions should be protected by patents, the Patent Office issued proposed guidelines<sup>46</sup> for its examiners on how to determine whether a claimed software-based invention was statutory subject matter. Following public comments, final guidelines<sup>47</sup> became effective on February 28, 1996. The Guidelines are the Patent Office's understanding of the various court decisions discussed previously and established procedures to be followed by patent examiners in the handling of patent applications for software-based inventions.

These Guidelines do not have the force of a Patent Office regulation, because Congress has given the Patent Office regulatory authority limited to the way proceedings are conducted in the Patent Office.

The theme of the Guidelines is that statutory subject matter for software-based inventions is based on the utility of the claimed invention.

The subject matter sought to be patented must be a “useful” process, machine, manufacture or composition of matter, *i.e.*, it must have a practical application. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of “real world” value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation

---

<sup>43</sup> 441 F.2d 682, 688, 169 USPQ 548, 553 (1971).

<sup>46</sup> 60 Fed. Reg. 28778 (1995).

<sup>47</sup> 61 Fed. Reg. 7478 (1996).

or research. Accordingly, a complete disclosure should contain some indication of the practical application for the claimed invention, *i.e.*, why the applicant believes the claimed invention is useful.

The utility of an invention must be within the “technological” arts. A computer-related invention is within the technological arts. A practical application of a computer-related invention is statutory subject matter. This requirement can be discerned from the variously phrased prohibitions against the patenting of abstract ideas, laws of nature or natural phenomena. An invention that has a practical application in the technological arts satisfies the utility requirement.<sup>48</sup> . . .

The subject matter courts have found to be outside the four statutory categories of invention is limited to abstract ideas, laws of nature and natural phenomena. While this is easily stated, determining whether an applicant is seeking to patent an abstract idea, a law of nature or a natural phenomenon has proven to be challenging. These three exclusions recognize that subject matter that is not *apactical application or use* of an idea, a law of nature or a natural phenomenon is not patentable.

Courts have expressed a concern over “preemption” of ideas, laws of nature or natural phenomena. The concern over preemption serves to bolster and justify the prohibition against the patenting of such subject matter. In fact, such concerns are only relevant to claiming a scientific truth or principle. Thus, a claim to an “abstract idea” is non-statutory because it does not represent a practical application of the idea, not because it would preempt the idea.<sup>49</sup>

The Guidelines give examples showing how the test of utility arguably reconciles all the past court decisions on when a software-based invention is statutory subject matter. They also make a procedural change that may substantially lower the number of statutory subject matter rejections – they require the examiner to address the novelty and nonobviousness of a claim even if it has been rejected as nonstatutory subject matter. A statutory subject matter rejection now results in additional work for the examiner and is not a quick way to dispose of the application.

### **III. Business Methods and *State Street Bank***

On July 23, 1998, the Federal Circuit made one of its clearest statements on the patentability of software-based inventions in its *State Street Bank*<sup>50</sup> decision. This was truly a long-awaited decision, since it wasn’t released until about 16 months after oral arguments were heard in the case.

Signature Financial had received a patent on “a data processing system for implementing an investment structure which was developed for use in Signature’s business as an administrator and accounting agent for mutual funds.” State Street Bank used a similar system, and after negotiations for a license to practice Signature’s

---

<sup>48</sup> 61 Fed. Reg. at 7480.

<sup>49</sup> 61 Fed. Reg. at 7481.

<sup>50</sup> *State Street Bank & Trust v. Signature Financial Group*, 149 F.3d 1368, 47 USPQ2d 1596 (Fed. Cir. 1998).

patent broke down, State Street Bank asked the district court to declare the patent invalid. The district court found the patent invalid because it did not claim statutory subject matter.

The original patent application had 12 claims – six method claims and six corresponding machine claims. When the examiner contemplated a statutory subject matter rejection of the method claims, Signature dropped them. The examiner then allowed the patent for the remaining machine claims. Claim 1 is representative of the machine claims, with the bracketed language indicating what the written description discloses as structure for the “mean for” limitations.

1. A data processing system for managing a financial services configuration of a portfolio established as a partnership, each partner being one of a plurality of funds, comprising:
  - (a) computer processor means [a personal computer including a CPU] for processing data;
  - (b) storage means [a data disk] for storing data on a storage medium;
  - (c) first means [an arithmetic logic circuit configured to prepare the data disk to magnetically store selected data] for initializing the storage medium;
  - (d) second means [an arithmetic logic circuit configured to retrieve information from a specific file, calculate incremental increases or decreases based on specific input, allocate the results on a percentage basis, and store the output in a separate file] for processing data regarding assets in the portfolio and each of the funds from a previous day and data regarding increases or decreases in each of the funds, [sic, funds] assets and for allocating the percentage share that each fund holds in the portfolio;
  - (e) third means [an arithmetic logic circuit configured to retrieve information from a specific file, calculate incremental increases and decreases based on specific input, allocate the results on a percentage basis and store the output in a separate file] for processing data regarding daily incremental income, expenses, and net realized gain or loss for the portfolio and for allocating such data among each fund;
  - (f) fourth means [an arithmetic logic circuit configured to retrieve information from a specific file, calculate incremental increases and decreases based on specific input, allocate the results on a percentage basis and store the output in a separate file] for processing data regarding daily net unrealized gain or loss for the portfolio and for allocating such data among each fund; and
  - (g) fifth means [an arithmetic logic circuit configured to retrieve information from specific files, calculate that information on an aggregate basis and store the output in a separate file] for processing data regarding aggregate year-end income, expenses, and capital gain or loss for the portfolio and each of the funds.

Clearly, the claim is to a specific machine, albeit one implemented using a conventional digital computer. But following *Alappat*, the programming of a general-purpose machine produces a special-purpose machine that performs the desired function. That does not end the analysis, though, because the district court found that

the machine fell into either the “mathematical algorithm” or “business method” exceptions to statutory subject matter.

The court first addressed the mathematical algorithm exception:

Unpatentable mathematical algorithms are identifiable by showing they are merely abstract ideas constituting disembodied concepts or truths that are not “useful.” From a practical standpoint, this means that to be patentable an algorithm must be applied in a “useful” way. In *Alappat*, we held that data, transformed by a machine through a series of mathematical calculations to produce a smooth waveform display on a rasterizer monitor, constituted a practical application of an abstract idea (a mathematical algorithm, formula, or calculation), because it produced “a useful, concrete and tangible result”—the smooth waveform.

Similarly, in *Arrhythmia Research*, we held that the transformation of electrocardiograph signals from a patient’s heartbeat by a machine through a series of mathematical calculations constituted a practical application of an abstract idea (a mathematical algorithm, formula, or calculation), because it corresponded to a useful, concrete or tangible thing—the condition of a patient’s heart.

Today, we hold that the 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, formula, 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.<sup>51</sup>

This reinforces the Guidelines’ theme of utility as the way to determine whether a claim is to statutory subject matter. Clearly, as long as the numbers being crunched have some meaning in the real world, the claimed invention is useful and statutory subject matter.

The court then addressed the viability of the *Freeman-Walter-Abele* test:

After *Diehr* and *Chakrabarty*, the *Freeman-Walter-Abele* test has little, if any, applicability to determining the presence of statutory subject matter. As we pointed out in *Alappat*, application of the 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. The test determines the presence of, for example, an algorithm. Under *Benson*, this may have been a sufficient indicium of nonstatutory subject matter. However, after *Diehr* and *Alappat*, 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 nonstatutory subject

---

<sup>51</sup> 149 F.3d at 1373, 47 USPQ2d at 1601 (citations omitted).

matter, unless, of course, its operation does not produce a “useful, concrete and tangible result.”<sup>52</sup>

The court then discussed the business method exception, taking “this opportunity to lay this ill-conceived exception to rest.” The decision of the district court was reversed and the case remanded to that court for further proceedings. In doing this, the court adopted the arguments first put forward by Judge Newman in her dissent in *In re Schrader*.<sup>53</sup>

When the *State Street Bank* decision was announced, many people (and the press) reported it as permitting a new type of patent – the business method patent. But the Patent Office had been granting patents of business techniques for many years. Some were for business machines, like Herman Hollerith’s punch-card handling machines (which formed the basis for IBM) and cash registers, but others were business methods like Signature’s. In 1982, patent 4,346,442 was issued to the brokerage firm Merrill Lynch, covering a “Securities Brokerage—Cash Management System.” Although another brokerage sued to have the patent declared invalid because it was for a business method, the district court<sup>54</sup> held that it was statutory subject matter. So not only were business method patents possible before *State Street Bank*, but one had been granted so long before *State Street Bank* that it was near its expiration at the time of that decision.

## IV. Other Ways of Claiming

## V. Printed Matter and Computer Software

*In re Lowry*<sup>64</sup> is about how a software invention whose novelty is in the way that it stores data can be claimed. The invention in *Lowry* was a “Data Processing System Having a Data Structure with a Single, Simple Primitive.” As noted in the decision, “The invention provides an efficient, flexible method of organizing stored data in a computer memory.” Claim 1 is representative.

A memory for storing data for access by an application program being executed on a data processing system, comprising:

adata structure stored in said memory, said data structure including information resident in a database used by said application program and including:

aplurality of attribute data objects stored in said memory, each of said attribute data objects containing different information from said database;

asingle holder attribute data object for each of said attribute data objects, each of said holder attribute data objects being one of said plurality of attribute data objects, a being-held relationship existing between each attribute data object and its holder attribute data object, and each of said attribute data objects having a being-held relationship with only a single other attribute data object, thereby establishing a hierarchy of said plurality of attribute data objects;

---

<sup>52</sup> 149 F.3d at 1374, 47 USPQ2d at 1601-1602 (citations omitted).

<sup>53</sup> 22 F.3d 290, 298, 30 USPQ2d 1455, 1462 (Fed. Cir. 1994) (Newman, J., dissenting).

<sup>54</sup> *Paine, Webber v. Merrill Lynch*, 564 F.Supp. 1358, 218 USPQ 212 (D. Del. 1983).

<sup>64</sup> 32 F.3d 1579, 32 USPQ2d 1031 (1994).



areferent attribute data object for at least one of said attribute data objects, said referent attribute data object being nonhierarchically related to a holder attribute data object for the same at least one of said attribute data objects and also being one of said plurality of attribute data objects, attribute data objects for which there exist only holder attribute data objects being called element data objects, and attribute data objects for which there also exist referent attribute data objects being called relation data objects; and

an apex data object stored in said memory and having no being-held relationship with any of said attribute data objects, however, at least one of said attribute data objects having a being-held relationship with said apex data object.

The examiner rejected this claim as nonstatutory under Section 101 and obvious under Section 103 in light of a prior art patent showing a different data structure stored in a computer memory. The Board reversed the examiner's rejection under Section 101 but affirmed the rejection under Section 103 under the printed matter doctrine of not giving weight to the printed matter when considering whether an invention is novel and nonobvious. As expressed in *In re Gulack*:

Where the printed matter is not functionally related to the substrate, the printed matter will not distinguish the invention from the prior art in terms of patentability.<sup>65</sup>

The Board held that Lowry had not shown that there was a functional relationship between the data structures and the memory containing them, and gave no weight to the nature of the data structures in the claims. Since the prior art patent disclosed a computer with memory containing data structures (although of a different form), disregarding the nature of Lowry's data structures makes his invention obvious in light of that patent.

Lowry appealed to the Federal Circuit. In its decision, the court first noted that "*Gulack* cautioned against a liberal use of 'printed matter rejections' under section 103" because it "stands on questionable legal and logical footing" and felt that it should not be extended to a new field, such as information stored in a computer memory.

This case, moreover, is distinguishable from the printed matter cases. The printed matter cases "dealt with claims defining as the invention certain novel arrangements of printed lines or characters, useful and intelligible only to the human mind." The printed matter cases have no factual relevance where "the invention as defined by the claims requires that the information be processed not by the mind but by a machine, the computer." Lowry's data structures, which according to Lowry greatly facilitate data management by data processing systems, are processed by a machine. Indeed, they are not accessible other than through sophisticated software systems. The printed matter cases have no factual relevance here. . . .

More than mere abstraction, the data structures are specific electrical or magnetic structural elements in a memory. According to Lowry, the data structures provide tangible benefits: data stored in

---

<sup>65</sup> 703 F.2d 1381, 1385, 217 USPQ 401, 404 (1983).

accordance with the claimed data structures are more easily accessed, stored, and erased. Lowry further notes that, unlike prior art data structures, Lowry's data structures simultaneously represent complex data accurately and enable powerful nested operations. In short, Lowry's data structures are physical entities that provide increased efficiency in computer operation. They are not analogous to printed matter. The Board is not at liberty to ignore such limitations.<sup>66</sup>

Since the elements of the claims limited to particular data structures could not be ignored, the claims were not obvious in light of prior art cited by the examiner, and the decision on the Board affirming the examiner was reversed.

But one should be careful about abolishing the printed matter doctrine too quickly, even when information is recorded in a way that can be perceived only by a machine. As Chief Judge Archer warned in his dissent in *Alappat*:

Consider for example the discovery or creation of music, a new song. Music of course is not patentable subject matter; a composer cannot obtain exclusive patent rights for the original creation of a musical composition. But now suppose the new melody is recorded on a compact disc. In such case, the particular musical composition will define an arrangement of minute pits in the surface of the compact disc material, and therefore will define its specific structure. Alternatively suppose the music is recorded on the rolls of a player piano or a music box.

Through the expedient of putting his music on known structure, can a composer now claim as his invention the structure of a compact disc or player piano roll containing the melody he discovered and obtain a patent therefor? The answer must be no. The composer admittedly has invented or discovered nothing but music. The discovery of music does not become patentable subject matter simply because there is an arbitrary claim to some structure.

And if a claim to a compact disc or piano roll containing a newly discovered song were regarded as a "manufacture" and within Section 101 simply because of the specific physical structure of the compact disc, the "practical effect" would be the granting of a patent for a discovery in music. Where the music is new, the precise structure of the disc or roll would be novel under Section 102. Because the patent law cannot examine music for "nonobviousness," the Patent and Trademark Office could not make a showing of obviousness under Section 103. The result would well be the award of a patent for the discovery of music.<sup>67</sup>

## V.A. Beauregard's Floppy Disks

*In re Beauregard* was an appeal by IBM of a decision of the Board of Patent Appeals that a floppy disk or other computer storage medium containing a novel and nonobvious computer program is not proper subject matter for a patent. (Because the

---

<sup>66</sup> 32 F.3d at 1583-1584, 32 USPQ2d at 1034-1035 (citations omitted).

<sup>67</sup> 33 F.3d 1526, 1553-1554, 31 USPQ2d 1545, 1565-1566 (1994) (Archer, C.J., dissenting) (citations omitted).

Patent Office agreed to Beauregard's arguments, there is no decision by the Federal Circuit, simply an order vacating the case and remanding it to the Patent Office.) Beauregard and his co-inventors, employees of IBM, had already received a patent (4,962,468) on a technique for filling the interior of a polygon displayed on a computer graphics device. As is common for software-based inventions, the patent claimed both methods for performing the filling technique and a system for performing the technique. The system could be either a general-purpose digital computer programmed to perform the method (the preferred embodiment) or a special-purpose graphics processor.

IBM was concerned that the primary people who would directly infringe these claims would be the end-users of graphics software, because it is only when the method is actually performed or when it configures a general-purpose computer that direct infringement occurs. Competitors would directly infringe the patent only when they ran the patented method, not when they made and sold floppy disks containing programs using the patented method. A competitor could be sued as a contributory infringer, but that generally means also suing a direct infringer (often a current or potential customer) and requires that the competitor using the patented method have actual knowledge of the patent. And if there is a substantial noninfringing use for the programs being sold, there may not be contributory infringement.

Since most software is distributed on a storage medium such as a floppy disk or CD-ROM, IBM figured that if it could get a patent on the storage medium containing the novel and nonobvious technique, it could sue the competitor as a direct infringer for making or selling the patented invention. In a continuing application, it claimed the invention as what it called a "computer program product" – a computer-readable medium having a program implementing the technique of the invention. In proceedings in the Patent Office, IBM argued that a computer program product was an article of manufacture or a component of a machine, both patentable subject matter.

The patent examiner and the Board of Patent Appeals rejected the claims as not being proper subject matter for a patent. They relied on a series of court decisions regarding "printed matter" which said that if the information stored on a medium was not related to that medium, the invention was not statutory or, alternatively, the novelty of the information could be disregarded when considering the novelty of a claim. Since there was nothing special about how the program was written on the floppy disk or other media, the Patent Office found the claims nonstatutory and obvious in light of other programs written on floppy disks.

IBM appealed the Patent Office's decision to the Federal Circuit. This was not surprising, since IBM had developed this case as a vehicle for the Federal Circuit to state whether computer program products were patentable subject matter. IBM was supported in its appeal by ten amicus briefs from industry and bar associations, stressing the importance of protecting computer software.

After seeing that there was virtually no popular support for its position, the Patent Office reversed its stand and declared that a computer program product was proper subject matter for a patent, that the printed matter doctrine did not apply to computer-readable programs, and that it was preparing appropriate guidelines on software patents for its examiners. The Federal Circuit found that there was no longer a dispute between the parties, and it remanded the application to the Patent Office for further prosecution in light of the Patent Office's new position.

## V.B. Patent Office Guidelines: Stored Information

Under the guidelines, the Patent Office will accept *Beauregard*-type claims to a computer program product. The proposed guidelines stated that “a computer-readable memory that can be used to direct a computer to function in a particular manner when used by the computer is a statutory ‘article of manufacture’.”<sup>68</sup> In fact, any memory device is an article of manufacture, whether its contents can direct the functioning of a computer or not.

Although the proposed guidelines included floppy disks and compact discs in the examples of a computer-readable medium that are statutory subject matter, disks storing a program, in contrast to a computer’s RAM, do not seem to fit the guideline’s own test. The bits stored in the program repository do not cause a computer to operate in a particular manner any more than any other data bits do. The program data on a disk must be first read and processed by a loader program to be placed in executable memory. It is only programs in RAM or ROM that cause a computer to perform the desired functions.

Furthermore, there is no difference in a floppy disk or compact disc itself if its stored information is a computer program that can be loaded into execution memory or data that can be converted to audio. Yet, according to the guidelines, one is a statutory article of manufacture and the other isn’t. The decision is based on the *intended use* of the data, not the *nature of* the possible article of manufacture. The same floppy disk, containing both a program and data operated on by a computer (such as text or digitally-encoded music or pictures) would be both an article of manufacture and not an article of manufacture, depending of what data on the floppy disk you consider.

It is important that law be anchored in reality, and distinctions be drawn only where they actually exist. The “mathematical algorithm” distinction did not work well because there was no similar distinction in computer science. In contrast, the “a general-purpose computer when programmed becomes a new machine” concept works well because the idea matches almost exactly the concept of virtual machines as discussed in operating systems.

The proposed guidelines made a distinction not well-anchored in computer science. They try say that a storage medium, or the bits it holds, is somehow different because of the intended use of a bit. By adding a qualification about its ultimate use, the guidelines depart from the intuitive concept that an article of manufacture is any tangible object made by man.

In response to public comments, the final guidelines took a different approach to drawing a line between patentable CD-ROMs holding computer programs and regular CDs holding music. It first noted that the statute requires an invention to be “useful”:

The subject matter sought to be patented must be a “useful” process, machine, manufacture or composition of matter, *i.e.*, it must have a practical application. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of “real world” value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation or research. Accordingly, a complete disclosure should contain some

---

<sup>68</sup> 60 Fed. Reg. 28778, 28778 (1995).

indication of the practical application for the claimed invention, *i.e.*, why the applicant believes the claimed invention is useful.

The utility of an invention must be within the “technological” arts. A computer-related invention is within the technological arts. A practical application of a computer-related invention is statutory subject matter. This requirement can be discerned from the variously phrased prohibitions against the patenting of abstract ideas, laws of nature or natural phenomena. An invention that has a practical application in the technological arts satisfies the utility requirement.<sup>69</sup>

The final guidelines then drew a distinction between function and nonfunctional descriptive material:

Descriptive material can be characterized as either “functional descriptive material” or “non-functional descriptive material.” In this context, “functional descriptive material” consists of data structures and computer programs which impart functionality when encoded on a computer-readable medium. “Non-functional descriptive material” includes but is not limited to music, literary works and a compilation or mere arrangement of data.

Both types of “descriptive material” are non-statutory when claimed as descriptive material *per se*. When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases. When non-functional descriptive material is recorded on some computer-readable medium, it is not structurally and functionally interrelated to the medium but is merely carried by the medium. Merely claiming non-functional descriptive material stored in a computer-readable medium does not make it statutory. Such a result would exalt form over substance. Thus, non-statutory music does not become statutory by merely recording it on a compact disk.<sup>70</sup>

It is likely that the validity of computer program product claims will not be tested in court until a software distributor is sued for infringement. The Patent Office now appears to allow any novel software-based invention that has any practical use, and will treat any media storing such an invention as a patentable article of manufacture.

## V.C. The Effect on Copyright

Although consideration should be given to the effect on copyright of patents for articles of manufacture that are programs stored in repository memories, such consideration was clearly outside the permissible scope of the PTO guidelines. Because programs are works of authorship stored in a tangible medium of expression, they are already protected by copyright. Further, copyright law not only provides protection to the author but also provides rights to users.

It is an infringement of a patent to make the patented invention. Whenever the program is copied, a new instance of the program stored in a memory is made. If a memory storing a particular program is patented, any copying results in a new

---

<sup>69</sup> 61 Fed. Reg. 7478, 7479 (1996).

<sup>70</sup> 61 Fed. Reg. at 7481.

instance of the program stored in memory that infringes the patent. For most inventions, a replicator capable of taking one instance of a machine or manufacture and directly producing a new instance would be something out of science fiction. For a program stored in memory, the copy command on the computer system is such a replicator.

For example, if the program was originally stored on a floppy disk, as would be the case when it is received from the software developer, another instance of the patented stored program is made when the program is copied from the floppy disk onto a hard disk. Another instance is made whenever an archive copy of the hard disk is made. Finally, each time the program is run it must be copied from the hard disk into the RAM of the computer, making yet another instance of the patented program stored in memory.

The patent laws do not contain an equivalent to Section 117 of the Copyright Act,<sup>71</sup> which permits copying of a computer program if it is a necessary step in its use. Anyone copying a program that includes a method that is the subject of a “program stored in memory” patent would infringe that patent, no matter how minor the method is to the entire computer program. This would be similar to the situation where an automobile containing a patented screw infringes the patent on that screw, even if the screw is an inconsequential part of the automobile.

In addition, patent law does not provide an equitable balance between the legitimate rights of the patent owner and the rights of users such as provided by “fair use” under Section 107 of the Copyright Act.<sup>72</sup>

It might be argued that the patent exhaustion doctrine, which allows the rightful owner of a patented device to use or sell that device, permits the making of new instances of patented stored programs if such copies are necessary for the use of the software. Nevertheless, the patent exhaustion doctrine has never allowed the owner of one instance of a patented article of manufacture to make another instance without the permission of the patent owner. On the contrary, there are many cases that try to resolve whether an apparatus is being repaired (not making the invention) or rebuilt (making the invention and infringing the patent). It is uncertain whether a court would extend the patent exclusion doctrine to include the making of new instances of stored programs.

Patents on computer programs residing in memory devices could preempt many of the user rights provided under the Copyright Act, particularly Section 117. By their nature, such patents will be infringed as part of the normal and expected usage of the programs. As a result, such patents distort both the patent and copyright systems simply to provide an avenue for suing the producer, instead of the end user, of a program that infringes a patented method.

## V.D. Beyond *Beauregard*

Unfortunately for the people who promoted *Beauregard* as a way of reaching the producer of infringing software without having to go through an end user, more and more software is being distributed not by floppy disk or CD-ROM but over the Internet. And that means there is no “article of manufacture” that stores the program being

---

<sup>71</sup> 17 U.S.C. §117.

<sup>72</sup> 17 U.S.C. §107.

distributed, and no clever way to claim the invention to reach the real infringer. While there is an article of manufacture produced on the end user's machine from the download, the whole purpose of *Beauregard* was to avoid having to look to the end user for any infringement.

In training materials for the Guidelines, the Patent Office suggested a possible claim to a propagated signal carrying a program. In its example, the program both compressed and encrypted data. This claim was:

- A computer data signal embodied in a carrier wave comprising:
- a. a compression source code segment comprising [the code]; and
  - b. an encryption source code segment comprising [the code].<sup>73</sup>

But is a data signal a “process, machine, manufacture, or composition of matter,” as required by Section 101.<sup>74</sup> If so, how far have we stretched the patentability boundary and what will be the consequences?

One justification for a *Beauregard*-type patent on a memory device, like a floppy disk, that stores a novel and nonobvious computer program is that the conventional ways of claiming a software-based invention did not provide proper protection against the true infringer. If the invention is claimed as a method, the claim is infringed only when the method is actually used. If the invention is claimed as a machine running the program, the claim is infringed only when the program is actually loaded into the computer. In both cases, it is the end user of the program who is the actual infringer, and who may not be aware that use of the software infringes a patent.

But the true infringer of the patent on the software-based invention is the competitor who included the patented method in its software and sold it to the end user. The competitor could be accused of inducing or contributing to the infringement of the patent. However, such an accusation presents a number of problems. For example, to be a contributory infringer, the competitor must know that what it is producing will lead to an infringement; they must have actual knowledge of the patent.

*Beauregard*-type claims attempt to get around these problems by claiming any memory device containing the patented software method. But as discussed above, such claims cause problems because end users infringe the patent whenever they copy or use the program, and because such claims base the patentability of a memory device on the intended use of its contents.

It is a red herring to say that it is more difficult to prove inducement of infringement or contributory infringement because some end user must be dragged into the case to show a direct infringement. First, it is highly unlikely that somebody is selling software infringing a patented method without testing it and thereby using the patented method, so such a software marketer is likely to be a direct infringer.

But more important, direct evidence of infringement by an end user is not required to prove inducement of infringement. Circumstantial evidence of infringement, such as evidence of extensive sales, is sufficient.<sup>75</sup>

---

<sup>73</sup> “Examination Guidelines For Computer-Related Inventions,” United States Patent and Trademark Office, 37, Claim 13 (1996).

<sup>74</sup> 35 U.S.C. §101.

<sup>75</sup> *Moleculon Research v. CBS*, 793 F.2d 1261, 1272, 229 USPQ 805, 813 (Fed. Cir. 1986).

It would be better to concentrate on getting court interpretations of Section 271<sup>76</sup> clearly holding that the distribution of a computer program whose use infringes a machine or process patent is inducing infringement or is contributory infringement, or congressional amendment of Section 271 to state it explicitly, rather than trying to make data signals an article of manufacture or to make floppy disks magically either an article of manufacture or not, depending on the intended use of the information on that floppy disk.

---

<sup>76</sup> 35 U.S.C. §271.