7 Tex. Intell. Prop. L.J. 225

Texas Intellectual Property Law Journal
Winter, 1999

SOFTWARE PATENTS

Wesley L. Austin[a1]

Table of Contents

*226 I. Introduction

Many years passed, and a number of conflicting court decisions were reached, before the courts concluded that software is indeed patentable subject matter. Some commentators adhere to the position that software should not be patentable.[1] Many of those against patenting software believe that the software industry will suffer because programmers will be unable to write software without infringing an existing software patent covering the programmer's design. These concerns are not new to the patent system; but they are relatively new to the software industry, and the popularity of software patents has increased dramatically.[2] Proponents of software patents believe that innovations in computer-implemented technologies should receive the same level of protection available to innovations in other technologies.[3] ***227** These advocates believe that requirements for novelty and non-obviousness to obtain a patent should address the concerns of those who think that software should not be patentable. Finally, a last group of commentators says that, while software should be protected, the existing system does not work well and should be changed.[4] Regardless of all these different voices, software patents are here to stay.[5]

Given that the argument about whether software should be patentable is currently resolved in favor of patentability, practitioners now struggle with writing, prosecuting, and enforcing software patents. Many articles published in the area of software patenting are mostly boring descriptions of insignificant cases with 300 pages of text before finally discussing the 1990s. In contrast, this article was written with the goal of providing tips to the practitioner for drafting software patent applications.

The author of this article assumes that readers are familiar with patent law, with computer programs, and with the different methodologies for their development. The scope of the article is limited to protecting software through the patent laws, although software can also be protected through trade secret and copyright laws as well.[6] In addition, the scope of the article is limited to subject matter, enablement, and best mode issues. The article does not address meeting the novelty and non-obviousness requirements for patentability.

Section II of this article discusses the case law leading up to the conclusion that software is patentable subject matter under 35 U.S.C. Section 101, analyzing those cases to determine the most important practices to ensure that a patent application meets subject matter requirements. Section III reviews the Patent and Trademark Office's Examination Guidelines for Computer-Related Inventions, and Section IV applies those guidelines and analyzes other recent literature to provide ***228** practical tips for writing patent claims for software inventions. Section V addresses writing the specification for a software patent application and suggests practices to ensure that the specification meets the requirements for enabling a person of ordinary skill in the art to practice the invention and for disclosing the best mode. Section VI presents the results of a case study of recently issued software patents, and Section VII concludes with a summary of the principles to follow when drafting software patent applications.

II. Subject Matter Case Law

Discussed in this section are the most well-known cases addressing proper subject matter for a patent. The case summaries focus on the important principles for which the cases are usually cited.[7] A practitioner should be familiar with the names of these cases and the key propositions for which they are cited, but need not be intimately familiar with the details and language of the cases.

A. Supreme Court Precedent

Throughout commentary written on software patents, several Supreme Court decisions are almost always cited: *Gottschalk v. Benson,*[8] *Parker v. Flook*,[9] and *Diamond v. Diehr*.[10]

1. *Gottschalk v. Benson*

The inventors in *Gottschalk* claimed "a method for converting binary-coded decimal (BCD) numerals into pure binary numerals."[11] Two claims, claim 8 and claim 13, were at issue.[12] The claims at issue were rejected by the Patent Office but sustained by the Court of Customs and Patent Appeals.[13] Claim 8 reads as follows:
***229** 8. The method of converting signals from binary coded decimal form into binary which comprises the steps of
(1) storing the binary coded decimal signals in a reentrant shift register,

(2) shifting the signals to the right by at least three places, until there is a binary '1' in the second position of said register,

(3) masking out said binary '1' in said second position of said register,

(4) adding a binary '1' to the first position of said register,

(5) shifting the signals to the left by two positions,

(6) adding a '1' to said first position, and

(7) shifting the signals to the right by at least three positions in preparation for a succeeding binary '1' in the second position of said register.[14]

In rejecting the claims as nonstatutory, Justice Douglas stated:

> It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting binary code to pure binary were patented in this case. The mathematical formula

> involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.[15]

The Court found that the claims were attempting to patent an abstract idea, reciting the "longstanding rule" that "an idea of itself is not patentable."[16] In addition, Justice Douglas stated that "[p]henomena of nature, though just discovered, mental processes, and abstract intellectual concepts are not patentable, as they are ***230** the basic tools of scientific and technological work."[17] He went on to say that "[i]f there is to be invention from such a discovery, it must come from the application of the law of nature to a new and useful end."[18] After laying the groundwork, Douglas went on to say that "the 'process' claim is so abstract and sweeping as to cover both known and unknown uses of the BCD to pure-binary conversion."[19] The Court was really finding that the claims were directed toward an abstract idea and thus were unpatentable. The Court's statement that "if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the *algorithm* itself"[20] was an unfortunate use of the term "algorithm."[21] It was unfortunate because all software can properly be referred to as different algorithms.[22] *Gottschalk* should not be read as a rule that "algorithms" are not patentable. In light of recent case law, algorithms are patentable to the extent that they are not abstract ideas.[23] Reading the court's language in *Gottschalk* concerning "algorithm" only confuses the issue. A better statement would have read, "if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the [abstract idea] itself."[24]

2. *Parker v. Flook*

The patent at issue in *Parker v. Flook* involved a method for updating alarm limits.[25] "The only novel feature of the method [was] a mathematical formula."[26] Conditions such as temperature, pressure, and flow rate are monitored during catalytic conversion processes.[27] When any of these conditions "exceeds a ***231** predetermined 'alarm limit,' an alarm may signal the presence of an abnormal condition indicating either inefficiency or perhaps danger."[28] During changing "operating situations, such as start-up, it may be necessary to 'update' the alarm limits periodically."[29]

The patent application in *Parker* describes a method of updating alarm limits where the "only difference between the conventional methods of changing alarm limits and that described … rests in … the mathematical algorithm or formula."[30] An operator can calculate an updated alarm limit using the formula "once he knows the original alarm base, the appropriate margin of safety, the time interval that should elapse between each updating, the current temperature (or other process variable), and the appropriate weighting factor to be used to average the original alarm base and the current temperature."[31]

The claim at issue, claim 1, stated:

1. A method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons wherein said alarm limit has a current value of

B_0+K

wherein B_0 is the current alarm base and K is a predetermined alarm offset which comprises:

(1) Determining the present value of said process variable, said present value being defined as PVL;

(2) Determining a new alarm base B_1, using the following equation:

$B_1=B_0(1.0-F)+PVL(F)$

where F is a predetermined number greater than zero and less than 1.0;

(3) Determining an updated alarm limit which is defined as B_1+K; and thereafter

(4) Adjusting said alarm limit to said updated alarm limit value.[32]

In analyzing the applicant's claim, the Court first assumed novelty and usefulness and that the applicant discovered the claimed invention.[33] The Court further assumed "that the formula is the only novel feature of respondent's ***232** method."[34] The main issue was "whether the discovery of this feature makes an otherwise conventional method eligible for patent protection."[35] In arguing for patentable subject matter, the respondent argued that his use of the formula would not "wholly preempt the mathematical formula" because there were other uses of the formula outside of the scope of his claims.[36] He also argued that the post-solution activity, "the adjustment of the alarm limit to the figure computed according to the formula," allows his process to be patentable in light of *Benson*.[37] The Court did not find these arguments persuasive.[38]

The Court quickly dismissed his argument of post-solution activity by stating that "[t]he notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance."[39] The Court pointed out that a good author "could attach some form of post-solution activity to almost any mathematical formula."[40] Justice Stevens stated that the process was unpatentable under Section 101 "not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention."[41]

The claim at issue did not tie itself into the environment as given in the claim preamble. In claim 1, the preamble recited a "method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons."[42] However, the elements of the claim merely recite a method for calculating a number without tying the claim into controlling something within that environment. This was the message Justice Stevens was trying to get across when he said "[a]n 'alarm limit' is a number."[43] The preamble language was merely an attempt to "limit the use of the ***233** formula to a particular technological environment."[44] In *Diamond v. Diehr*, the Court stated that the principle that one cannot patent a formula in the abstract "cannot be circumvented by attempting to limit the use of the formula to a particular technological environment."[45]

Not only did the claim not tie itself into the environment, but the specification did little to help. The court pointed out that "[t]he patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables."[46] Moreover, Justice Stevens further stated "[n]or does [the specification] purport to contain any disclosure relating to the chemical processes at work, the monitoring of process variables, or the means of setting off an alarm or adjusting an alarm system."[47] The claim simply states an algorithm for updating an alarm limit.[48] Again, this seems to emphasize that all the patent application did was show one how to calculate a number. From this case, an author learns to always draft a specification containing a full disclosure of the invention, the environment in which it operates, and how the invention works in that specific environment.[49] "When a mathematical formula is involved, be sure to disclose how to measure the variables and how to select appropriate values of any constants, multiplier factors, and so forth."[50]

3. *Diamond v. Diehr*

In *Diamond v. Diehr*, the Supreme Court clearly opened the door for software patents.[51] The invention in *Diehr* involved "a process for molding raw, uncured synthetic rubber into cured precision products."[52] The Arrhenius equation, which reflects well-known time, temperature, and cure relationships, enables one to calculate when to open the press and remove the cured product.[53] However, the ***234** respondents asserted that it is difficult to perform the necessary computations to determine cure time "because the temperature of the molding press could not be precisely measured," whereby the industry has not been able to obtain uniformly accurate cures.[54] Accordingly, sometimes this led to instances of "overestimating the mold-opening time and overcuring the rubber," and in other instances this led to "underestimating that time and undercuring the product."[55]

Respondents' process constantly measures the actual temperature inside the mold and automatically feeds these measurements to a computer.[56] The computer iteratively recalculates the cure time by use of the Arrhenius equation.[57] The computer continues to recalculate the cure time until the cure time equals the actual time that has elapsed since the press was closed.[58] Then the computer signals a device to open the press.[59] "According to the respondents, the continuous measuring of the temperature inside the mold cavity, the feeding of this information to a digital computer which constantly recalculates the cure time, and the signaling by the computer to open the press, are all new in the art."[60]

Diehr and Lutton's application contained 11 different claims.[61] Claim 1 is as follows:
***235** 1. A method of operating a rubber-molding press for precision molded compounds with the aid of a digital computer, comprising:

providing said computer with a data base for said press including at least,

natural logarithm conversion data (ln),

the activation energy constant (C) unique to each batch of said compound being molded, and

a constant (x) dependent upon the geometry of the particular mold of the press,

initiating an interval timer in said computer upon the closure of the press for monitoring the elapsed time of said closure,

constantly determining the temperature (Z) of the mold at a location closely adjacent to the mold cavity in the press during molding,

constantly providing the computer with the temperature (Z),

repetitively calculating in the computer, at frequent intervals during each cure, the Arrhenius equation for reaction time during the cure, which is
$\ln v = CZ+x$
where v is the total required cure time,

repetitively comparing in the computer at said frequent intervals during the cure each said calculation of the total required cure time calculated with the Arrhenius equation and said elapsed time, and

***236** opening the press automatically when a said comparison indicates equivalence.[62]

Relying on *Gottschalk*, the patent examiner rejected the claims as nonstatutory subject matter under 35 U. S. C. Section 101.[63] The Patent and Trademark Office Board of Appeals agreed with the examiner.[64] However, the Court of Customs and Patent Appeals reversed.[65]

In beginning its statutory subject matter analysis, the Court stated that "Congress intended statutory subject matter to 'include anything under the sun that is made by man.'"[66] The Court further recited, in defining the nature of a patentable process, that a process is "an act, or a series of acts, performed upon the subject-matter to be transformed and reduced to a different state or thing."[67] Justice Rehnquist then stated that in *Gottschalk*, the Court had repeated the definition recited in *Cochrane v. Deener*,[68] adding: "Transformation and reduction of an article 'to a different state or thing' is the clue to the patentability of a process claim that does not include particular machines."[69] From these statements, the Court concluded that "we think that a physical and chemical process for molding precision synthetic rubber products falls within the § 101 categories of possibly patentable subject matter."[70] "That respondents' claims involve the transformation of an article, in this case raw, uncured synthetic rubber, into a different state or thing cannot be disputed."[71]

The Court articulated that the patentability of the claims was "not altered by the fact that in several steps of the process a mathematical equation and a programmed digital computer are used."[72] Justice Rehnquist then enumerated the ***237** categories of nonstatutory subject matter as "laws of nature, natural phenomena, and abstract ideas."[73] He then stated that the "holdings in *Gottschalk v. Benson* … and *Parker v. Flook* … both of which are computer-related, stand for no more than these long-established principles."[74]

The Court compared the claims at issue to the claims in *Parker* stating that "[i]n contrast [to the claims in *Parker*], the respondents here do not seek to patent a mathematical formula."[75] Although the process employed a well-known mathematical equation, the claims did not "pre-empt the use of that equation."[76] "Rather, they seek only to foreclose from others the use of that equation in conjunction with all of the other steps in their claimed process."[77] To help point out that the claims were not merely claiming an abstract idea and enumerating steps for calculating some number, in the abstract, the Court enumerated the real-world, physical steps involved in the claims as including "installing rubber in a press, closing the mold, constantly determining the temperature of the mold, constantly recalculating the appropriate cure time through the use of the formula and a digital computer, and automatically opening the press at the proper time."[78]

Moreover, the process does not become unpatentable simply because it is accomplished through a computer.[79] "Our earlier opinions lend support to our present conclusion 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."[80] One key phrase in the last line is "simply because it uses." The mathematical formula and/or computer program must be used to accomplish something more than solving an equation or calculating a number. Thus used, the formula and/or program are "simply being used" to accomplish some specific, real task. "It is now commonplace that an *application* of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection."[81] Again, the Court is stating that the formula and/or computer program must be used to do something.
***238** [W]hen a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (e. g., transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of § 101.[82]

4. Synthesis

From the preceding Supreme Court cases we have several broad principles. "Phenomena 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."[83] From *Parker v. Flook* and *Diamond v. Diehr*, we learn that claiming a method for simply calculating a number, without any ties to the real world, will not meet the statutory requirements, and adding a few terms in the preamble and tacking on illusory post-solution activity won't save it.[84] In addition, the specification should show the practical application of the invention.[85] A patent specification that does not tie claims to the real world and the invention's practical applications may not be patentable.[86] Finally, *Diamond v. Diehr* indicates that software is patentable subject matter.[87] *Diehr* teaches that transformation and reduction of an article to a different state or thing is the clue to the patentability of a method claim.[88]

Gottschalk, Parker, and *Diehr* are best used as references for the preceding principles without analyzing the language of the cases in detail. When read carefully and critically analyzed in light of recent case law from the Federal Circuit, these cases may cause confusion.[89] The software industry is still undergoing a technological revolution,[90] and the language of cases decided over 20 years ago is increasingly obsolete. Know the Supreme Court cases for the broad principles set forth in this article, and for more specificity in software patents, rely on recent case law from the Federal Circuit.

*239 B. Federal Circuit Case Law

In addition to the Supreme Court cases discussed in the previous section, the following cases shed light on how the Federal Circuit construes claims and how it reads the foregoing Supreme Court decisions decided many years ago.

1. *In re Alappat*

The Federal Circuit's decision in *In re Alappat*[91] illustrates important principles to be followed in claiming a software invention. The invention in *Alappat* related "generally to a means for creating a smooth waveform display in a digital oscilloscope."[92] Because of the limitations of the cathode-ray tube (CRT) and the limitations of the digital processing occurring within the oscilloscope, the waveform displayed on the oscilloscope may exhibit discontinuity, jaggedness, or oscillation.[93] To overcome this shortcoming, the invention in *Alappat* implemented an anti-aliasing system where the pixel illumination was modulated such that the intensity of each pixel's illumination decreased as its distance from the real waveform increased.[94] "Employing this *anti-aliasing* technique eliminate [[[d] any apparent discontinuity, jaggedness, or oscillation in the waveform, *thus giving the visual appearance of a smooth continuous waveform*."[95]

The only independent claim at issue in *Alappat* was claim 15, which reads:
A rasterizer for converting vector list data representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:
(a) means for determining the vertical distance between the endpoints of each of the vectors in the data list;

(b) means for determining the elevation of a row of pixels that is spanned by the vector;

(c) means for normalizing the vertical distance and elevation; and

(d) means for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.[96]

***240** The dependent claims at issue, claims 16-19, each depended "directly from claim 15 and more specifically define[d] an element of the rasterizer claimed therein."[97] The Examiner, under 35 U.S.C. Section 101, issued a final rejection of claims 15-19 "'because the claimed invention is nonstatutory subject matter," and the original three-member Board panel reversed this rejection.[98] However, in a reconsideration decision, a five member majority of the Board panel affirmed the Examiner's Section 101 rejection. The majority held that the PTO need not apply Section 112 para. 6 in rendering patentability determinations.[99] In commenting on the need to apply Section 112 para. 6 in patentability determinations, the *Alappat* court stated that the majority "characteriz[ed] this court's statements to the contrary in *In re Iwahashi* 'as dicta,' and dismiss[ed] this court's discussion of Section 112 para. 6 in *Arrhythmia Research Technology, Inc. v. Corazonix Corp.* on the basis that the rules of claim construction in infringement actions differ from the rules for claim interpretation during prosecution in the PTO."[100]

The majority of the Board panel "held that each of the means recited in claim 15 reads on any and every means for performing the particular function recited."[101] The Board panel found that the claim was directed to nothing more than a mathematical algorithm and concluded that the claim was directed to nonstatutory subject matter.[102]

The court in *Alappat* pointed out that the "Board majority … erred as a matter of law in refusing to apply Section 112 para. 6 in rendering its § 101 patentable subject matter determination."[103] The majority in *Alappat* then recited independent claim 15 replacing the means clauses with the corresponding structure which Alappat disclosed in his specification and also adding the words "or an equivalent thereof."[104] The "Section 112 para. 6 enhanced claim" read as follows:
A rasterizer [a "machine"] for converting vector list data representing sample magnitudes of an input waveform into anti-aliased pixel illumination intensity data to be displayed on a display means comprising:
***241** (a) [an arithmetic logic <u>circuit</u> configured to perform an absolute value function, or an equivalent thereof] for determining the vertical distance between the endpoints of each of the vectors in the data list;

(b) [an arithmetic logic <u>circuit</u> configured to perform an absolute value function, or an equivalent thereof] for determining the elevation of a row of pixels that is spanned by the vector;

(c) [a pair of barrel <u>shifters</u>, or equivalents thereof] for normalizing the vertical distance and elevation; and

(d) [a <u>read only memory</u> (ROM) containing illumination intensity data, or an equivalent thereof] for outputting illumination intensity data as a predetermined function of the normalized vertical distance and elevation.[105]

The court then stated "[a]s is evident, claim 15 unquestionably recites a machine, or apparatus, made up of a combination of known electronic circuitry elements" and "[b]ecause claim 15 is directed to a 'machine,' which is one of the four categories of patentable subject matter enumerated in § 101, claim 15 appears on its face to be directed to § 101 subject matter."[106]

The *Alappat* court went on to address the Board majority's argument that the claimed subject matter was within a "judicially created exception to Section 101 which the majority refers to as the 'mathematical algorithm' exception."[107] Citing *Diehr*, the court stated that there are "three categories of subject matter for which one may not obtain patent protection, namely 'laws of nature, natural phenomena, and abstract ideas.'"[108] Judge Rich concluded that the Supreme Court never intended to exclude a broad category of subject matter, mathematical algorithms, from Section 101.[109] "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."[110]

Judge Rich set forth the proper inquiry regarding "the so called mathematical subject matter exception to § 101" as focusing on "whether the claimed subject matter as a *whole* is a disembodied mathematical concept … which in essence represents nothing more than a 'law of nature,' 'natural phenomenon,' or 'abstract ***242** idea.' If so, *Diehr* precludes the patenting of that subject matter."[111] Turning to the facts of the case, the court stated that the invention in *Alappat* was not an abstract idea, "but rather a specific machine to produce a useful, concrete, and tangible result."[112] "[C]laim 15 is limited to the use of a particularly claimed combination of elements performing the particularly claimed combination of calculations to transform, i.e., rasterize, digitized waveforms (data) into anti-aliased, pixel illumination data to produce a smooth waveform."[113]

The preamble of claim 15 aided the court in finding that the claimed subject matter was for a specific purpose and not just an abstract idea. The preamble was not a "not a mere field-of-use label having no significance. Indeed, the preamble specifically recites that the claimed rasterizer converts waveform data into output illumination data for a display, and the means elements recited in the body of the claim make reference not only to the inputted waveform data recited in the preamble but also to the output illumination data also recited in the preamble."[114] Judge Rich then stated that "[c]laim 15 thus defines a combination of elements constituting a machine for producing an anti-aliased waveform."[115]

Finally, the court in *Alappat* addressed the Board majority's reasoning that claim 15 was unpatentable merely because it "reads on a general purpose digital computer 'means' to perform the various steps under program control."[116] Just because claim 15 reads on a "general purpose computer programmed to carry out the claimed invention" does not mean that the claim is unpatentable as directed to nonstatutory subject matter.[117] This is because "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."[118] Following the Board majority's reasoning, "a programmed general purpose computer could never be viewed as patentable subject matter under § 101."[119] "This reasoning is without basis in the law."[120]

243 2. *In re Warmerdam

The invention in *In re Warmerdam* was directed to a "method and apparatus for controlling the motion of objects and machines, such as robotic machines, to avoid collision with other moving or fixed objects."[121] The object is treated as if it were surrounded by a bubble of sufficient size to enclose the object.[122] It is then assumed that any motion that bursts the bubble would produce a collision.[123] The invention in *Warmerdam* claimed an improvement upon the prior art bubble systems. The improvement was in the inventors' "bubble hierarchy." The essence of the improvement was that upon collision detection with a bubble, the invention in *Warmerdam* would replace the bubble with a set of smaller, more refined bubbles.[124] Thereafter, another assessment would ascertain whether a collision was still going to occur, even with the refined estimate of the object's occupied area.[125] Appellants "bubble hierarchy" could iterate and define increasingly better safety zones.[126] The iteration of increasingly more refined bubble estimates would "repeat[] itself until it is determined either that (1) the anticipated path does not intersect any of the bubbles at a particular level of the hierarchy, indicating collision avoidance, or (2) the anticipated path intersects one of the bubbles at the lowest level of the hierarchy, indicating that a collision will occur."[127]

The claimed invention in *Warmerdam* "includes methods for generating a 'data structure'—undefined as such but presumably including the measured dimensions and coordinates of the bubble hierarchy—and a machine (presumably a general purpose computer) having a memory containing data representing a bubble hierarchy as generated by any of the claimed methods."[128] The novelty in Warmerdam's invention was asserted to be in "the generation and placement of the hierarchy of bubbles along the medial axis of the object."[129] The specification of Warmerdam defined the medial axis of an object "to be 'a line with the same topology as the ***244** object itself connecting points which lie midway between boundary centers of the object."'[130]

Six claims were at issue in the appeal of Warmerdam. Claims 1 through 4 were method claims; claim 5 was an apparatus claim; and claim 6 was directed toward a data structure. The claims were as follows:
1. A method for generating a data structure which represents the shape of [[[[sic] physical object in a position and/or motion control machine as a hierarchy of bubbles, comprising the steps of:
first locating the medial axis of the object and

then creating a hierarchy of bubbles on the medial axis.

2. The method of Claim 1 wherein the step of creating the hierarchy comprises a top-down procedure of:
first placing a root bubble which is centered at the center of gravity of the object and has a radius equal to the maximum distance from the center of gravity to the contour of the object;

next, if the medial axis has a plurality of branch lines, placing a plurality of first successive bubbles each of which encompasses a distinct part of the object which is described by one of said branch lines; and

then successively dividing each line of the medial axis into two new line parts and placing a pair of next successive bubbles each of which encompasses a distinct part of the object which is described by one of said new line parts.

3. The method of Claim 1 wherein the step of creating the hierarchy comprises a bottom-up procedure of:
first representing the medial axis as [sic] large plurality of discrete points;

next placing the centers of a plurality of lowest level bubbles at said discrete points, where the radius of each bubble is equal to the minimum

distance from the corresponding center point to the contour of the object; and

then successively creating new bubbles by merging the smallest bubble remaining with its smallest neighbor(s) to create a new bubble and repeating

this step until only one root bubble remains.

4. The method of Claim 3 wherein two old bubbles are merged to yield a new bubble in accordance with the formulas:
$r' = (r1+j+r2)/2$,

$x' = 1/2(x1+x2+(r1-(x1-x2))/(r2/j))$

$y' = 1/2(y1+y2+(r1-(y1-y2))/(r2/j))$

***245** $z' = 1/2(z1+z2+(r1-(z1-z2))/(r2/j))$;
wherein r1 and r2 are the radii of the old bubbles, j is the distance between the centers of the old bubbles (x1, y1, z1) and (x2, y2, z2) are the coordinates of the center of the old bubbles, r' is the radius of the new bubbles, and (x', y', z') are the coordinates of the center of the new bubble.

5. A machine having a memory which contains data representing a bubble hierarchy generated by the method of any of Claims 1 through 4.

6. A data structure generated by the method of any of Claims 1 through 4.[131]

Claims 1-4 and 6 were rejected by the examiner for lack of statutory subject matter under 35 U.S.C. Section 101, while claim 5 was rejected for indefiniteness under 35 U.S.C. Section 112 para. 2.[132] After a final rejection, Warmerdam appealed to the Board.[133] The Board sustained the rejections reasoning that claims 1-4 "recited no more than a mathematical algorithm in the abstract, and thus failed to comply with § 101," that claim 5 was "indefinite under § 112 para. 2 because it left 'unclear and unexplained how a memory is made or produced by the steps of generating recited in claims 1 through 4,"' and reasoning that claim 6 "failed to satisfy § 101 on the ground that a 'data structure' is not within one of the categories of patentable subject matter listed in § 101, to wit, a process, machine, manufacture, composition of matter, or improvements thereof."[134] Warmerdam appealed the Board's decision.

The court in *Warmerdam* first stated that, rather than relying on the *Freeman*[135]-*Walter*[136]-*Abele*[137] test, the court would return to the "language of the statute and the Supreme Court's basic principles as enunciated in *Diehr*, and eschew efforts to describe nonstatutory subject matter in other terms."[138] The *Freeman-* ***246** *Walter-Abele* test was a two step protocol wherein "the first step … is to determine whether a mathematical algorithm is recited directly or indirectly in the claim, and the second step … is to determine whether the claimed invention as a whole is no more than the algorithm itself."[139] Judge Plager stated that the problem with this test is "that there is no clear agreement as to what [] a 'mathematical algorithm' [is], which makes rather dicey the determination of whether the claim as a whole is no more than that."[140]

The court sustained the rejections of claims 1-4 under Section 101.[141] The court reasoned:
The body of claim 1 recites the steps of "locating" a medial axis, and "creating" a bubble hierarchy. These steps describe nothing more than the manipulation of basic mathematical constructs, the paradigmatic "abstract idea." As the Supreme Court has made clear, "an idea of itself is not patentable," *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. 498, 507 (1874); taking several abstract ideas and manipulating them together adds nothing to the basic equation.[142]

The court reversed the Board's determination that claim 5 was indefinite stating that "[c]laim 5 is for a machine, and is clearly patentable subject matter."[143] The court recited the legal standard for definiteness as being "whether a claim reasonably apprises those of skill in the art of its scope."[144] Claims 1-4 were invalid because they were nothing more than the manipulation of abstract ideas. The constructs recited in claims 1-4 were not tied to anything. All claim 5 added was a simple binding between these abstract ideas and a "machine having a memory."[145] This simple addition, giving context to these abstract ideas and claiming them as being implemented on a machine having a memory was enough for the court to state that claim 5 was "*clearly* patentable subject matter."[146]

***247** The court sustained the rejection of claim 6 for lack of statutory subject matter.[147] Claim 6 did not tie itself to any physical machine or memory; it seemed to claim a data structure "in the air." The court stated that "the 'data structure' of claim 6 is nothing more than another way of describing the manipulation of ideas contained in claims 1-4, it suffers from the same fatal defect they do."[148] It is interesting to note that claim 6 is essentially claim 5 without the binding to a machine and memory.[149] Claim 5 recited "[a] machine having a memory which contains data representing a bubble hierarchy generated by the method of any of Claims 1 through 4," while claim 6 recited a "[a] data structure generated by the method of any of Claims 1 through 4."[150] Claim 5 essentially claims the same data structure as claim 6. The difference is that claim 5 ties the data structure to a machine and memory while claim 6 claims the data structure in the abstract.

3. *In re Lowry*

Similar to *Warmerdam, Lowry* involved a patent application relating to "the storage, use, and management of information residing in a memory."[151] Claim 1 is representative of claims 1-5 that claim a memory containing a stored data structure.[152] Claim 1 read as follows:
1. A memory for storing data for access by an application program being executed on a data processing system, comprising:
a data structure stored in said memory, said data structure including information resident in a database used by said application program and including:

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

a single 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;

a referent 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 ***248** 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.[153]

Claims 6 through 29 were also at issue in *Lowry*.[154] "Claims 6 through 19 claim a data processing system executing an application program, containing a database, a central processing unit (CPU) means for processing the application program, and a memory means for holding the claimed data structure."[155] Claims 20 through 29 were method claims for accessing, creating, adding, and erasing the data structure and/or portions thereof.[156]

The examiner rejected claims 1 through 5 under 35 U.S.C. Section 101 as nonstatutory subject matter.[157] In addition, the examiner rejected claims 1 through 19 under 35 U.S.C. Section 103,[158] and claims 20 through 29 under 35 U.S.C. Section 102(e).[159]

The Board reversed the 35 U.S.C. Section 101 rejection finding that "the invention claimed in claims 1 through 5 was statutory subject matter."[160] The Board affirmed the rejection of claims 1 through 19 under 35 U.S.C. Section 103 and affirmed the rejection of claims 20 through 29 under 35 U.S.C. Section 102(e).[161] The Board "analogized Lowry's data structure … to printed matter and relied on … *In re Gulack*."[162] As a result, when "evaluating patentability under Sections 102 and 103, the Board failed to give patentable weight to the claimed data structure."[163]

***249** The Federal Circuit corrected the Board on their printed matter rejection stating that "[t]he 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.'"[164] The court reiterated that "Lowry's data structures … are processed by a machine" and are only accessible "through sophisticated software systems."[165] "The printed matter cases have no factual relevance here. Nor are the data structures analogous to printed matter."[166]

The court in *Lowry* went on to state that "Lowry does not claim merely the information content of a memory."[167] The court looked to the physical organization of the claimed data structures at issue in *Lowry* stating that "the claims require specific electronic structural elements which impart a physical organization on the information stored in memory."[168] The court went on to say, "[i]ndeed, Lowry does not seek to patent the Attributive data model in the abstract. Rather, Lowry's data structures impose a physical organization on the data."[169] The court continued to emphasize the physicality of the data structures in Lowry by summarizing that "[i]n 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."[170]

After pointing out that the data structures in *Lowry* deserved patentable weight, the court went on to reverse the Board's finding that claims 1 through 19 were obvious.[171] The court also reversed the Board's determination that claims 20 through 29 were anticipated.[172]

*250 4. *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*

In *State Street*, the Federal Circuit sent a clear message that software patents are here to stay.[173] The patent at issue in *State Street*, U.S. Patent No. 5,193,056, was directed toward a data processing system "for monitoring and recording the information flow and data, and making all calculations, necessary for maintaining a partnership portfolio and partner fund (Hub and Spoke) financial services configuration."[174] In summarizing the invention, the Federal Circuit stated that "[i]n essence, the system … facilitates a structure whereby mutual funds (Spokes) pool their assets in an investment portfolio (Hub) organized as a partnership. This investment configuration provides the administrator of a mutual fund with the advantageous combination of economies of scale in administering investments coupled with the tax advantages of a partnership."[175]

Claim 1 of the '056 patent reads as follows:
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 for processing data;

(b) storage means for storing data on a storage medium;

(c) first means for initializing the storage medium;

(d) second means 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, assets and for allocating the percentage share that each fund holds in the portfolio;

(e) third means 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 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 for processing data regarding aggregate year-end income, expenses, and capital gain or loss for the portfolio and each of the funds.[176]

***251** State Street and Signature Financial are both agents for multi-tiered partnership fund financial services.[177] State Street pursued a license from Signature to use the invention of the ‘056 patent.[178] Negotiations between the parties failed and State Street filed a declaratory judgment action asserting invalidity, unenforceability, and noninfringement.[179] State Street then filed a partial motion for summary judgment.[180] The trial court granted the motion of patent invalidity for failure to claim statutory subject matter, and Signature Financial appealed.[181]

The Federal Circuit reversed and remanded.[182] In reversing, the Federal Circuit held 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.”[183]

The Federal Circuit made it extremely difficult to find a claimed algorithm unpatentable simply because it is a mathematical algorithm. Judge Rich stated that “[u]npatentable 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.”[184] So, to clear any mathematical algorithm hurdles, a software patent claim must be embodied and useful.[185] By definition, software is an embodiment of an algorithm. Once someone has written the code, they have embodied the algorithm. With regard to the utility requirements of Section 101, a claimed software invention should easily be shown to be useful.

The *Freeman-Walter-Abele* test is no longer viable.[186] The Federal Circuit stated that it was error for the trial court to have applied the *Freeman-Walter-Abele* ***252** test “to determine whether the claimed subject matter was an unpatentable abstract idea.”[187] “After *Diehr* and *Chakrabarty*, the *Freeman-Walter-Abele* test has little, if any, applicability to determining the presence of statutory subject matter.”[188]

Finally, in *State Street* the Federal Circuit put the “business method exception” out of its misery. “We take this opportunity to lay this ill-conceived exception to rest.”[189] This exception had never been used by the Federal Circuit or by the C.C.P.A. to find an invention unpatentable.[190] Before *State Street*, although no one wanted to retire the “business method exception”, no one wanted to use it as the basis for invalidating a patent either. Now that this exception has been expressly rejected by the Federal Circuit, patent applicants should not hesitate to file patent claims that claim a business method.

5. Synthesis

The above cases teach us several points about subject matter patentability for software. One must apply Section 112 para. 6 to means-plus-function claim elements when determining patentability.[191]

Several principles have been enunciated regarding mathematical algorithms. No fourth category of subject matter is excluded from Section 101 under the label of a mathematical algorithm.[192] A disembodied mathematical concept is nonstatutory because it represents nothing more than a ‘law of nature,’ ‘natural phenomenon,’ or ‘abstract idea.’[193] Certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas and will not be patentable until reduced to some type of practical application.[194] However, after *State Street*, all that is required to overcome this hurdle is to have the claimed invention be embodied in a computer program and show that it is “useful.”[195]

***253** Claim preambles should be tied to the real-world environment in which the invention will operate.[196] Claim elements should be linked to the real-world environment and to the claim preamble to show the practical application of the invention.[197]

Software effectively changes a general purpose computer into a new machine: “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.”[198]

The printed matter cases have no factual relevance to claims to data structures.[199] When claiming data structures, emphasize the physical organization that is required by the data structures; do not claim the data structures in the abstract where any conceivable physical organization would read on the claim.[200] Since a “business method exception” to statutory subject matter no longer exists, business method claims cannot be rejected or invalidated on that basis.

III. Examination Guidelines for Computer-Related Inventions

The Patent and Trademark Office issued Examination Guidelines for Computer-Related Inventions[201] (Guidelines) to “assist Office personnel in the examination of applications drawn to computer-related inventions.”[202] The Guidelines are not the law, they do not constitute substantive rulemaking, and the failure of Office personnel to follow the guidelines is neither appealable nor petitionable.[203] Of course, the PTO believes that the Guidelines are fully consistent with the rulings of the Supreme Court and the Federal Circuit.[204]

Because patents issued by the PTO “carry a presumption of validity that is often difficult to challenge,” the Guidelines may, as a practical matter, have the ***254** effect of law.[205] Moreover, the decision by the Federal Circuit in *In re Trovato*[206] underscores the importance of the Guidelines. Trovato applied for a patent on a computer-implemented invention.[207] The Patent Examiner rejected the claims as being unpatentable subject matter under Section 101.[208] The Board upheld the nonstatutory rejection of most of the claims.[209] The Federal Circuit also agreed with the Examiner and denied Trovato’s patent in a decision in 1994.[210] The Federal Circuit later withdrew its opinion, vacated the decisions of the Board, and remanded to the PTO for reconsideration in light of *Alappat* and “any new guidelines adopted by the Patent and Trademark Office for examination of computer-implemented inventions.”[211] In dissent, Judge Nies said that the Guidelines “must yield to precedent from this court and the Supreme Court” concerning § 101, and that it appeared “that the majority [was] provid[ing] an advisory opinion endorsing the proposed guidelines.”[212] Nevertheless, the Guidelines should be followed in drafting claims to software inventions.

A. Determine What Applicant Has Invented and Is Seeking to Patent

According to the Guidelines, Office personnel should begin by reviewing the complete specification, including the detailed description of the invention, the claims, any specific embodiments that have been disclosed, and any specific utilities that have been asserted for the invention.[213] Examiners “will no longer begin examination by determining if a claim recites a ‘mathematical algorithm.”’[214]

***255** Rather than focusing on whether a claim recites a mathematical algorithm, an Examiner will determine if the invention has a practical application and

possess[es] 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 indication of the practical application for the claimed invention, i.e., why the applicant believes the claimed invention is useful.[215]

The M.P.E.P concludes that “[a] practical application of a computer-related invention is statutory subject matter.”[216]

The Guidelines instruct Examiners to begin evaluating a computer-related invention by "determin[ing] what the programmed computer does when it performs the processes dictated by the software."[217] In addition, Office personnel are to "determine how the computer is to be configured to provide that functionality (i.e., what elements constitute the programmed computer and how those elements are configured and interrelated to provide the specified functionality)."[218] If applicable, Examiners should "determine the relationship of the programmed computer to other subject matter outside the computer that constitutes the invention."[219]

The Guidelines instruct Office personnel to begin "claim analysis by identifying and evaluating each claim limitation."[220] Claim limitations in process claims "will define steps or acts to be performed."[221] In product claims that are directed to either machines, manufactures or compositions of matter, the claim limitations "will define discrete physical structures."[222] "The discrete physical structures may be comprised of hardware or a combination of hardware and software."[223] With all claims, whether or not they use means-plus-function language, examiners are to correlate each claim limitation to all portions of the disclosure describing that claim limitation to ensure that the claim is correctly interpreted.[224]

***256** Certain terms in a claim, such as "whereby" and "adapted for," may not limit the scope of a claim. "Language that suggests or makes optional but does not require steps to be performed or does not limit a claim to a particular structure does not limit the scope of a claim or claim limitation."[225] The following examples of language that may not limit a claim are given in the M.P.E.P.: "(A) statements of intended use or field of use, (B) 'adapted to' or 'adapted for' clauses, (C) 'wherein' clauses, or (D) 'whereby' clauses."[226]

The applicant does not have to provide a disclosure that describes elements of his or her invention that are well known in the art.[227] Such elements "will be construed as encompassing any and every art—recognized hardware or combination of hardware and software technique for implementing the defined requisite functionalities."[228] "Office personnel are to give claims their broadest reasonable interpretation in light of the supporting disclosure."[229] Means-plus-function language in claims "must be interpreted to read on only the structures or materials disclosed in the specification and 'equivalents thereof.'"[230]

B. Determine Whether the Claimed Invention Complies with 35 U.S.C. Section 101

The M.P.E.P. states that 35 U.S.C. Section 101 "defines four categories of inventions that Congress deemed to be the appropriate subject matter of a patent; namely, processes, machines, manufactures and compositions of matter."[231] "Subject matter not within one of the four statutory invention categories or which is not 'useful' in a patent sense is, accordingly, not eligible to be patented."[232] The Guidelines further point out that subject matter outside of these four statutory categories is limited to "abstract ideas, laws of nature and natural phenomena."[233] "These three exclusions recognize that subject matter that is not a practical ***257** application or use of an idea, a law of nature or a natural phenomenon is not patentable."[234]

The Guidelines instruct Office personnel to classify "each claim into one or more statutory or nonstatutory categories."[235]

1. Nonstatutory Subject Matter

Nonstatutory subject matter falls into the categories of natural phenomena, abstract ideas, and laws of nature.[236] Claims to computer-related inventions that are nonstatutory under the categories of abstract ideas or laws of nature constitute "descriptive material."[237] "Descriptive material can be characterized as either 'functional descriptive material' or 'nonfunctional descriptive material.'"[238] Functional descriptive material "consists of data structures and computer programs which impart functionality when encoded on a computer-readable medium."[239] Nonfunctional descriptive material[240] "includes but is not limited to music, literary works and a compilation or mere arrangement of data."[241] The Guidelines state that "[b]oth types of 'descriptive material' are nonstatutory when claimed as descriptive material *per se*."[242] However, "[w]hen 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."[243]

*258 a. Functional Descriptive Material: "Data Structures" Representing Descriptive Material *Per se* or Computer Programs Representing Computer Listings *Per se*

Claimed data structures encoded on computer-readable medium are statutory because the claim defines "structural and functional interrelationships between the data structure and the medium which permit the data structure's functionality to be realized ...."[244] Computer programs are treated similarly to data structures.[245] Computer programs claimed simply as computer listings[246] *per se* are not physical things and are not statutory.[247] However, like data structures, "a claimed computer-readable medium encoded with a computer program defines structural and functional interrelationships between the computer program and the medium which permit the computer program's functionality to be realized, and is thus statutory."[248] A claim of an otherwise statutory manufacture or machine remains statutory even if a computer program is included in the claim.[249] Likewise, where a computer program is used as part of a process claim where the instructions of the computer program are being executed by a computer, the claim is statutory regardless of the inclusion of the computer program.[250] "Only when the claimed invention taken as a whole is directed to a mere program listing … is it descriptive material *per se* and hence nonstatutory."[251] A "computer program itself is not a process and Office personnel should treat a claim for a computer program, without the computer-readable medium needed to realize the computer program's functionality, as nonstatutory functional descriptive material."[252]

*259 2. Statutory Subject Matter

a. Statutory Product Claims

Statutory product claims may be either machines, manufactures or compositions of matter.[253] A software invention claimed as a product will either be a claim for a machine or for a manufacture. "If a claim defines a useful machine or manufacture by identifying the physical structure of the machine or manufacture in terms of its hardware or hardware and software combination, it defines a statutory product."[254] A claim for a machine or manufacture will be treated as either (1) a claim that encompasses any and every machine or manufacture for performing the underlying process, or (2) a claim that "defines a specific machine or manufacture."[255]

Office personnel are to examine a claim of the first type (i.e., encompassing any and every machine or manufacture for performing the underlying process) by evaluating "the underlying process the computer will perform in order to determine the patentability of the product."[256] The specification will determine whether the product claim encompasses any and every computer implementation of a particular process.[257] The M.P.E.P. enumerates two characteristics of such claims: (1) the claim will "define the physical characteristics of a computer or computer component exclusively as functions or steps to be performed on or by a computer," and (2) the claim will "encompass any and every product in the stated class (e.g., computer, computer-readable memory) configured in any manner to perform that process."[258]

If a claim does not encompass every computer implementation of a process, "then it must be treated as a specific machine or manufacture."[259] "Generally a claim drawn to a particular programmed computer should identify the elements of the computer and indicate how those elements are configured in either hardware or a combination of hardware and specific software."[260] The M.P.E.P. also states that to ***260** "adequately define a specific computer memory, the claim must identify a general or specific memory and the specific software which provides the functionality stored in the memory."[261]

If the disclosure, when describing elements in the software process claim, states that "it would be a matter of routine skill to select an appropriate … computer system … [to] implement the claimed process," and does not describe specific software or hardware to accomplish the task, the claim would encompass any computer embodiment of that process claim.[262] On the other hand, if the specification describes specific software that is to be used to configure a general purpose computer, the claim defines a specific computer because the disclosure identifies the "specific machine capable of performing the indicated functions."[263]

b. Statutory Process Claims

The Guidelines state that for a computer-related process claim to be statutory, the claim "must either: (A) result in a physical transformation outside the computer for which a practical application in the technological arts is either disclosed in the specification or [is known in the art], or (B) be limited by the language in the claim to a practical application within the technological arts."[264]

A claim will be found to be statutory if it results in a physical transformation outside the computer.[265] The M.P.E.P. refers to specific categories ("safe harbors") within which a claimed process will be found to have resulted in a physical transformation outside the computer.[266] The safe harbors are "independent physical acts (post-computer process activity)" and "manipulation of data representing physical objects or activities (pre-computer process activity)."[267]

The physical transformation in the post-computer process activity must be more than merely "conveying the direct result of the computer operation."[268] The independent physical acts of post-computer process activity will make the process statutory if the acts "involve the manipulation of tangible physical objects and result ***261** in the object having a different physical attribute or structure."[269] Examples of this type of statutory process include a method of curing rubber in a mold involving several process steps and ending in "opening the mold."[270] Another example is a method of controlling a mechanical robot ending in "controlling the robot's movement and position based on [a] calculated position."[271]

The pre-computer process activity safe harbor also is clearly statutory.[272] The pre-computer process activity safe harbor "requires the measurements of physical objects or activities to be transformed outside of the computer into computer data, where the data comprises signals corresponding to physical objects or activities external to the computer system, and where the process causes a physical transformation of the signals which are intangible representations of the physical objects or activities."[273] Several examples were listed in the M.P.E.P. of statutory processes of this type. The examples included a method to analyze "electrical signals and data representative of human cardiac activity;" a method for receiving, processing, and displaying CATscan images of a patient; and a method to conduct seismic exploration "by imparting spherical seismic energy waves into the earth from a seismic source, generating a plurality of reflected signals in response to the seismic energy waves at a set of receiver positions in an array, and summing the reflection signals to produce a signal simulating the reflection response of the earth to the seismic energy."[274] After each example given, the M.P.E.P. pointed out the "real world value" of each process.[275] If a claimed process has no real world value, the claim will most likely not be statutory.

A claimed process will be found statutory if it falls within one of the safe harbors enumerated by the M.P.E.P., i.e., independent physical acts (post-computer process activity) or manipulation of data representing physical objects or activities (pre-computer process activity). However, even if a claim does not clearly fall under one or both of the safe harbors, "the claim may still be statutory if it is limited by the language in the claim to a practical application in the technological arts."[276]

***262** A process will be nonstatutory if it "merely manipulates an abstract idea or performs a purely mathematical algorithm," regardless of whether the process is useful or not.[277] For this kind of subject matter to be statutory, "the claimed process must be limited to a practical application of the abstract idea or mathematical algorithm in the technological arts."[278] Examples given of claimed processes limited to a practical application were as follows: "[a] computerized method of optimally controlling transfer, storage and retrieval of data between cache and hard disk storage devices such that the most frequently used data is readily available;" "[a] method of controlling parallel processors to accomplish multi-tasking of several computing tasks to maximize computing efficiency;" "[a] method of making a word processor by … executing [a] stored program to impart word processing functionality to the general purpose digital computer;" and "[a] digital filtering process for removing noise from a digital signal comprising the steps of calculating a mathematical algorithm to produce a correction signal and subtracting the correction signal from the digital signal to remove the noise."[279]

Claims not meeting the standards mentioned (safe harbors or practical application) will likely be found nonstatutory. Claims will be found to define nonstatutory processes if they "consist solely of mathematical operations without some claimed practical application," or "simply manipulate abstract ideas … without some claimed practical application."[280] "[W]hen a claim reciting a mathematical algorithm is found to define nonstatutory subject matter the basis of the … rejection must be that, when taken as a whole, the claim recites a law of nature, a natural phenomenon, or an abstract idea."[281]

c. Certain Claim Language Related to Mathematical Operation Steps of a Process

An author may add clauses reciting a practical application to a claim hoping that the examiner will find patentable subject matter. Simply placing this type of "context clause" in the claim, especially only in the claim preamble, will have little, if any, effect. "Claim language that simply specifies an intended use or field of use for the invention generally will not limit be scope of a claim, particularly when only presented in the claim preamble."[282]

***263** In certain circumstances, steps of "collecting" or "selecting" data for use in a process comprising one or more mathematical operations will not act as a limitation on the claim.[283] The antecedent acts will not act to limit the claim if they are dictated by nothing other "than the performance of a mathematical operation."[284] If the process represents a practical application of one or more mathematical operations, acts performed to create the data that will be used in the process will be treated as further limiting the claim.[285]

Similar to the foregoing antecedent steps, in some situations, certain post-solution activity will not further limit a process claim.[286] If a post-mathematical operation "represent[s] some 'significant use' of the solution, [[[it] will invariably impose an independent limitation on the claim."[287] The M.P.E.P. defines "significant use" as "any activity which is more than merely outputting the direct result of the mathematical operation."[288] The Guidelines gave several examples of steps found not to independently limit a process claim involving at least one mathematical operation step. The examples of steps not independently limiting included a step of updating alarm limits,[289] a step of magnetically recording the result of a calculation,[290] and a step of transmitting electrical signals representing the result of calculations.[291]

C. Evaluate Application for Compliance with 35 U.S.C. Section 112

Claims must particularly point out and distinctly claim the invention.[292] The claims, in light of the specification, must "reasonably apprise a person of ordinary skill in the art of the invention."[293] Every feature or element of the invention need not be explicitly recited.[294] "In fact, it is preferable for claims to be drafted in a form ***264** that emphasizes what the applicant has invented (i.e., what is new rather than old)."[295]

For means-plus-function language to distinctly claim an invention, the detailed description must "make[] it clear that the means corresponds to well-defined structure of a computer or computer component implemented in either hardware or software and its associated hardware platform."[296] Unless structures or materials corresponding to a means-plus-function limitation are well known in the art, the specification must disclose these structures or materials or the claim will fail to particularly point out and distinctly claim the invention.[297]

The specification of a patent must enable a person skilled in the art to make and use the claimed invention without undue experimentation.[298] Even if experimentation is not simple, it may not be undue. "The fact that experimentation is complex, however, will not make it undue if a person of skill in the art typically engages in such complex experimentation."[299] This statement may be particularly true of software patents involving very large and complicated computer programs interacting over a network. The foregoing statements seem to concede that for such a complex system, a person skilled in the art would expect to perform complex experimentation to get the invention working properly even with an enabling disclosure.

A software patent specification should teach one of ordinary skill in the art how to configure a computer to implement the invention. "The specification should disclose <u>how</u> to configure a computer to possess the requisite functionality or <u>how</u> to integrate the programmed computer with other elements of the invention, unless a skilled artisan would know how to do so without such disclosure."[300] The Guidelines state that "[i]n many instances, an applicant will describe a programmed computer by outlining the significant elements of the programmed computer using a functional block diagram."[301] In most instances, the blocks in a block diagram are labeled with some indication of the block's function. The M.P.E.P. suggests to examiners that they "should review the specification to ensure that along with the functional block diagram the disclosure provides information that adequately ***265** describes each 'element' in hardware or hardware and its associated software and how such elements are interrelated."[302]

D. Synthesis

Several important principles are emphasized by the Guidelines. Computerrelated inventions "must have a practical application."[303] Inventions must "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."[304] In determining patentability, nonstatutory subject matter falls into the categories of natural phenomena, abstract ideas, and laws of nature.[305] Claims to computer-related inventions that are nonstatutory under the categories of abstract ideas or laws of nature constitute "descriptive material," either functional or nonfunctional descriptive material.[306] The Guidelines state that "[b]oth types of 'descriptive material' are nonstatutory when claimed as descriptive material *per se*."[307] However, "[w]hen 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."[308]

A process will be found to be statutory if it results in a physical transformation outside the computer.[309] The Guidelines refer to specific categories (safe harbors) within which a claimed process will be found to have resulted in a physical transformation outside the computer.[310] The safe harbors are “independent physical acts (post-computer process activity)” and “manipulation of data representing physical objects or activities (pre-computer process activity).”[311]

***266 IV. Practical Tips in Writing Claims**

Before drafting claims for an invention, the most important task for the author is to understand what the invention is and how it works. Claim drafters should not be in such a hurry to draft claims that they skip the step of understanding the technology and the invention. Diving into claim drafting before understanding the invention is like diving into writing an exam answer without reading the entire question.

A patent attorney should also be aware of the prior art, enabling him or her to distinguish the invention from the prior art. In fact, the author must really understand the state of the prior art to ascertain what the invention is, and what it is not. Understanding the invention also includes understanding what the presently preferred embodiments are, and what the best mode is.

Several important questions should be asked of the inventor: How will the invention be packaged? Will it be packaged and/or distributed with other pieces of software and/or hardware? Are the customers of this product consumers or other manufacturers?[312] The claims to be drafted will be affected by the answers to these questions. For example, if an invention is targeted at consumers, to be distributed on CD-ROMs, a few claims directed toward the CD-ROM (or a more generally termed computer-readable medium), the functional data stored thereon, and the interrelationships between the data should be included. On the other hand, if the software to be patented is only to be distributed with a specific piece of hardware (e.g., the software that runs on a processor in a VCR), and it is not going to be distributed on a storage medium by itself, such a data structure claim to a storage medium may be unnecessary.

In drafting claims for software, remember to tie the invention to the real world. Make it clear to the Examiner, judge, and jury the invention’s real-world application. If the claims read like a mathematical formula, they probably do not have enough material supporting a practical application of the invention. “To improve the likelihood that a claim covering a program or a program-implemented method or system will be considered statutory, draft the claim with an ‘atmosphere’ of computer implementation rather than of programming *per se*.”[313]

***267 A. Draft the Claims First**

The detailed description of a patent must provide support for the claims. Therefore, at least some of the broad claims should be written before the detailed description is drafted. Some authors prefer to draft the detailed description before drafting the claims.[314] This school believes that the understanding gained from drafting the specification helps them to better understand the invention. When it comes time to draft claims, unless an author uses all the same terms used in the specification, a portion of the specification must be rewritten to provide support for the claims.[315] If the only way that a practitioner can understand the invention is to write a detailed description of it, then it may be appropriate to write the detailed description first. This strategy should be undertaken with the understanding that much of the specification may need rewriting after the claims are completed because the author will have developed a better understanding of the invention while drafting claims.[316]

Despite the competing practice of writing the specification first, the preferred method is to draft the claims first.[317] Drafting the claims before the specification distinctly points out the invention to the author thereby telling him or her exactly what must be supported in the specification.[318] “Draft the specification after all the claims are finalized. The claims form a detailed outline of the specification.”[319] Of course, while writing the detailed description, and adding detail not found in the claims, other important features may become apparent requiring additional claims to protect those features.[320] Accordingly, an author will likely add several dependent claims while drafting the detailed description.[321]

***268** A useful tool to use while drafting claims is to draw claim diagrams to aid in understanding the invention. Claim diagrams usually consist of several circles or boxes connected by lines. The circles/boxes represent the claimed elements, and the lines interconnecting them represent the claimed relationships between the elements. Claim diagrams are simple to make from an existing claim. Each element or sub-element is drawn as a circle or box and labeled. Each relationship between elements or sub-elements is drawn as a line between the related elements and labeled with a name indicating the type of relationship.[322] Claim diagrams are beneficial because the reader can “see” what is being claimed. When claims are completed, the author also has a rough draft of the sketches needed for the patent application.

The claim set should include claims of varying scope: from broad claims to picture claims.[323] The broad claims should include only those parts that are essential to the invention.[324] The claims of intermediate and narrow scopes should include the most likely commercial embodiments of the invention, including non-essential elements more as the claims get narrower. "A picture claim is important because it often will be allowed in a first office action, uncontaminated by file wrapper estoppel. Such a claim may prove to be significant during litigation."[325]

The claim set should include several independent claims, of varying scope and disparate approaches.[326] Drafting claims using different approaches means that a patent application for a piece of software typically will have both apparatus and method claims.[327]

Building an entire set of claims depending from a single independent claim is unwise because if there is a problem with that independent claim, all the claims may have problems. In addition, putting 'all the claim eggs into one basket' can result in a failure to claim other important subcombinations of features that cannot depend ***269** from that particular independent claim.[328] "Thus, additional independent claims using different language or a different approach to describing the combination should be considered to assure full coverage of the scope of the invention."[329]

B. Method Claims

A patent author can use several different approaches in claiming software. "Most computer program-related invention claims have been drafted either as a process describing a set of actions to be performed on or by specific combinations of means-plus-function elements, or as a component of a new machine, in order to satisfy the requirement of Section 101."[330] Method claims will be discussed first, followed by apparatus claims.

Method claims are defined in terms of operations performed. Because method claims are intended to define the steps performed and not the hardware, they need not be tied as closely to the disclosed structure as the apparatus claims should be.[331] Where the hardware is not unique, method claims should dominate or at least not be shortchanged.[332] For example, if a computer program runs on any IBM PC compatible computer running Microsoft Windows 98, the method claims should dominate the claim set because novelty is in the computer program alone—the steps performed to achieve a desired result.[333]

A method claim usually protects the steps performed, very similar to a computer program, with a minimum of structural limitations found in the claim. On the other hand, an apparatus claim is usually tied much closer to a piece of hardware. Thus, a method claim is more capable of cleanly claiming and protecting a computer program without unnecessarily limiting its application to a particular piece ***270** of hardware. However, to ensure that a method claim is proper subject matter, a wise author will tie it to some piece of hardware.[334]

C. Apparatus Claims

Apparatus claims are more often found to be statutory than method claims.[335] Most of the software patents analyzed in the study described in Section VI of this article included at least one apparatus claim. Furthermore, almost all of the apparatus claims included means-plus-function language.[336] This seems to be consistent with the apparent preference for method claims. *In re Alappat* clearly put the Federal Circuit's stamp of approval on means-plus-function language in software patents.[337]

If method claims have already been drafted for the invention, drafting corresponding apparatus claims is relatively simple and can be done by simply prefacing each step with "means for."[338] A benefit of this approach is that it precludes a restriction requirement between the apparatus and method claims.[339] Remember that because a wise patent author always includes varying claims of different scope, at least one apparatus claim set should be free from any means-plus-function language.

1. Means-Plus-Function Claims

As stated, if the method claims are drafted first, one can arrive at apparatus claims rather quickly. Using the foregoing method does mean that each element of the apparatus claim will be in means-plus-function language. Some have suggested ***271** that at least one element of the apparatus claim be in structural language (not means-plus-function) to help the claim meet the requirements for patentable subject matter.[340] Other commentators have suggested drafting one set of apparatus claims with means-plus-function elements and another set with hardware component elements.[341]

In drafting means-plus-function language, use the "means for" language authorized by 35 U.S.C. Section 112.[342] That statute states:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.[343]

According to the law, the means-plus-function element will be construed to cover the structure disclosed in the specification and equivalents thereof. In a case of infringement, an element will infringe a means-plus-function element if it is an equivalent to the element disclosed in the specification. If it is the same or an equivalent, literal infringement will be found.[344] If the element in the claim was not a means-plus-function element, then literal infringement will not be found unless the potentially infringing element is the same structure as claimed. Thus, means-plus-function language significantly aids in finding literal infringement because it covers the structure in the specification and equivalents thereof. In contrast, language not in means-plus-function form only covers the structure in the specification.

Some commentators recommend using broad functional language rather than means-plus-function terminology, particularly when claiming a very broad claim element.[345] The argument for using a broad functional name such as 'processor' ***272** instead of a "means for" element is that using "means for processing" might actually limit that element more. "Means for processing" would cover the structure disclosed in the specification and equivalent structures, whereas "processor" may be interpreted to cover any "processor."[346] A counterargument is that a means-plus-function element will cover the structures disclosed and their structural equivalents. If a broad, functional, non-means term covers additional structures, then it must cover not only the structures disclosed in the specification and their structural equivalents, but also non-structural equivalents (or structural non-equivalents). Convincing an examiner or court to interpret the claims in this way seems unlikely.

Of course, broad functional terms, like "processor", may be construed by Examiners and courts as falling within the scope of Section 112 para. 6.[347] In addition, an author using the term "means for processing," could simply describe the "means for processing" in the specification as being a "processor."

Taking under consideration that at least one element should use non-means-plus-function language, the author should choose an element having a broad functional name (like processor, memory, input device, etc.) for that element. The elements in the claim that are truly novel — those that are not in the prior art — should be the means-plus-function elements.

To ensure that an invention is adequately protected from competitors, the invention should be claimed in different ways with varying scope. Using several different approaches in claiming an invention, of varying degree of breadth, casts a broader and stronger net to catch infringers. Because it is desirable to include claims of varying scope, claims with and without means-plus-function should be included with the patent application.[348]

D. Data Structure Apparatus Claims

A data structure claim is a modification of an apparatus claim directed toward some aspect of the data embodied on a computer-readable medium (e.g., a CDROM, ***273** hard drive, memory, and the like). *Warmerdam* and *Lowry* both included data structure claims. These types of claims are infrequently used because they are often narrower than a method or apparatus claim for the same invention. Unless a data structure claim is written carefully, or is really a method or apparatus claim in disguise,[349] designing around the claim to avoid infringement is easy.

A data structure claim is less likely to be considered patentable than an apparatus claim.[350] "[I]f the data structure claims have some physical mapping to real world physical objects, and especially to resulting structure within the computer, then they are more likely to survive."[351]

Given that apparatus and method claims will read on those instances where that apparatus or method is being practiced, data structure claims are primarily used for software inventions to ensure that manufacturers of infringing products can be reached as direct infringers.

Software-related inventions often reside within a software product, namely, a memory device (e.g., floppy diskette, CD-ROM) embodying machine readable program instructions (code). Such instructions direct a computer system to perform useful, new and non-obvious tasks. Most software vendors who manufacture and market memory devices embodying program code do not manufacture and/or market the computer system that executes the program code. Accordingly, traditional apparatus and method claims that positively recite a computer system may not protect software product inventions from direct patent infringement. Rather, direct infringement occurs only by a consumer who purchases an infringing memory device from the software vendor and incorporates it into a machine. Consequently, the patent owner's legal remedies against the software vendor may be limited to contributory infringement. However, for contributory infringement, legal remedies require additional elements of proof and can be complicated by defensive maneuvers, including substantial non-infringing uses.[352]

Claim drafters should include at least one claim directed toward what would typically be sold in the marketplace. That is, at least one set of claims should be directed toward the functional-descriptive material recorded on a computer-readable medium.[353] This claim set would be written to cover a company that is producing the CD-ROMs with software that infringes the patent on direct infringement, not just contributory infringement.

***274 V. The Specification**

From the case law and the PTO Guidelines, as discussed earlier in Sections II and III, the more ties the description has to the real world, the more favorably a court will look upon the patent.[354] Therefore, an author should tie the invention in the real world by clearly showing how pieces of his invention interface with physical objects that the readers can relate to initially, or that they can relate to after a quick perusal of the background section.

The law requires the patent specification to meet the enablement and best mode requirements. These requirements are set forth as follows:

> 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 our his invention.[355]

A. Enablement

In order to meet the enablement requirement, a patent specification must enable one skilled in the art to make and use the invention.[356] Who is one skilled in the art? "When the challenged subject matter is a computer program that implements a claimed device or method, enablement is determined from the viewpoint of a skilled programmer using the knowledge and skill with which such a person is charged."[357] The nature of the invention, the role the program plays in the invention, and the complexity of the computer program all influence the amount of disclosure required.[358]

Many computer-related inventions involve other arts (e.g., heart signals, seismic signals, curing rubber, and the like). When the invention involves different ***275** arts, the specification is adequate if it enables one skilled in each art to carry out the aspect of the invention in their art.[359]

Where the invention relates to one field, e.g., seismic signals, and also involves a computer program, the requisite disclosure as to those elements relating to seismic signals is judged by the level of skill in the seismic arts, while the sufficiency of the disclosure required for the program part of the invention depends on the level of skill in programming.[360]

Because the enablement requirement is measured by one skilled in the art, the specification must disclose enough so that only "routine programming efforts" remain to be accomplished.[361] While some experimentation may be necessary to implement the invention, a need for "undue experimentation" will fail the enablement requirement. In short, to meet the enablement requirement, a patent must disclose enough for a programmer of ordinary skill to create and use it without undue experimentation.[362]

There are several different means commonly used to describe computer programs. These means can vary in detail from a simple textual description of what function(s) the computer program accomplishes to a listing of the computer program in a programming language. Other means include flow charts, block diagrams, data flow diagrams, structure charts, object diagrams,[363] event trace diagrams, state diagrams, pseudocode, etc.[364] All of these are not needed to meet the enablement requirement.

At a minimum, to meet the enablement requirement a patent should include a detailed description of the computer program, the function(s) it accomplishes, and the environment it operates in, in such detail that a skilled programmer could ***276** implement the invention without undue experimentation.[365] A commentator has said that this description "should have about as much detail as would be equivalent to a flow chart in which the lowest level block is well known to skilled programmers or could be derived by such a programmer without undue experimentation."[366]

In this detailed description, the invention should not be described in terms of a specific programming language, but rather generally disclosed in terms not tied to a specific programming language.[367] Describing the invention entirely in "C"[368] terms, for example, may limit any means-plus-function language in the claims to specific "C" function calls, and equivalents thereof. For example, assume that the patent includes a claim including a means-plus-function element. Further assume that this means-plus-function element is basically a module to be executed by a computer, and that in the description this means is described in terms of specific "C" function calls. If other languages do not have functional equivalents to these "C" calls for the means-plus-function element, that element may be limited to the specific "C" function calls. However, this problem could easily be remedied by, in the detailed description, first describing the task to be accomplished in general terms, and then giving examples, in at least two or more programming languages, what function calls could be used to accomplish the task. A goal to strive for in drafting a patent is that any readers of the patent should understand that the author contemplated using almost any programming language available to implement the invention—that the invention is not to be limited to a specific programming language.

A good author will include flow charts with his patent application.[369] Flow charts are helpful because usually they step through what is happening at a high enough level to impart some understanding to the readers. Flow charts are much easier to understand and follow than a program listing or pseudocode. Usually flow charts are in general terms and not in terms of specific programming languages, and accordingly, will not limit the scope of means-plus-function elements.[370] The written ***277** description of the patent should describe each box, decision, transition, etc., in the flow chart in sufficient detail that a skilled programmer could implement each item without undue experimentation.[371]

Source code is usually not needed for enablement.[372] A skilled programmer can write routines to accomplish specific tasks; he or she does not need to be given the source code to do it. If the patent must enable one of no skill in the art to implement a computer related invention, the source code may be required. However, if the person of ordinary skill in the art is a skilled programmer, the source code is not usually needed. Source code does little to help one understand how a piece of software works.[373] Another possible reason to exclude source code with the patent application is the possibility of providing code that doesn't work. If the source code does not compile, or it compiles and links but doesn't run properly, the enablement and/or best mode requirements may be in jeopardy. "A complete program listing in the specification in support of a complex computer program-implemented without accompanying explanatory text may be difficult or impossible to understand, and may not be enabling if there are important errors in the listing."[374] If one skilled in the art could make the invention, then the enablement requirement is met. However, if opposing counsel is trying to invalidate a patent, source code that doesn't work gives him or her powerful ammunition to use.

Despite these problems, some circumstances justify including at least a portion of the invention's source code. If a part of the invention involves something a bit tricky, unorthodox, or just different enough that one of ordinary skill in the art would have to perform undue experimentation to get it to work, the author should include ***278** an example of such source code with the patent.[375] However, if flow charts are included with a software patent application, seldom would the author need to include source code.[376] Source code would only be needed when a skilled programmer could not implement the invention without undue experimentation.[377]

For software, at least one figure should be included describing the hardware environment in which the software invention will run. If the program runs on a typical personal computer, one block diagram will likely suffice. In fact, it may not even be necessary to include a hardware block diagram if the invention is aimed at any PC because the tools used by programmers in the PC environment insulate the programmer from the hardware. However, to support apparatus claims, at least one block diagram of the hardware to be used should be included. If the invention is aimed at a less well-known hardware platform (e.g., a cellular phone, a VCR, a fax machine), more than one hardware block diagram may be necessary to sufficiently disclose the context of the invention.

The blocks in the block diagrams should correspond to well-known hardware.[378] In the specification, unless it would be clear to one skilled in the relevant art, the author should identify specific parts that may be used for the blocks in the block diagram.[379] For example, it should not be necessary, for a computer program aimed at any PC, for the author to state specific model numbers and manufacturers for the different types of memory that may be used with the hardware. For a portion of these types of computer program inventions, those implementing ***279** the invention would never need to know such things. However, in other areas, where the development tools are not so advanced and well known, it would be wise for the author to identify specific parts that may be used in combination with his or her invention.

For those blocks not commercially available and known in the art, another figure should be included to explain the inner workings of this block, and the detailed description should explain how this part works.[380] One skilled in the art must be able to make and use the invention; therefore, care should be taken to assure that those skilled in the art will understand how blocks not well-known in the art can be implemented.

B. Best Mode

A patent specification must set forth the best mode contemplated by the inventor of carrying our his invention.[381]
Determining whether a patent complies with the best mode requirement involves two underlying factual inquiries. First, it must be determined whether, at the time the patent application was filed, the inventor had a best mode of practicing the claimed invention.... Second, if the inventor had a best mode of practicing the claimed invention, it must be determined whether the specification adequately disclosed what the inventor contemplated as the best mode so that those having ordinary skill in the art could practice it.[382]

Depending upon the size of the software system and the number of people working on it, a complex piece of software can change from hour to hour. On large software projects, each night the team may "build" a new version of the software. Of course, some software inventions are directed toward smaller aspects of software systems, rather than the whole system. But, inventors do try to claim large software systems. The description should attempt to satisfy the best mode requirement at a level higher than source code to avoid a problem with dislosing an outdated version of the source code. If the author does include source code with the application, and the source code is not the most recent version as of the filing date, the patent may not meet the best mode requirement. For example, on Oct. 1 a patent attorney files a patent application including the source code for program ABC, with a version ***280** 1.100.[383] Later, when the patent is litigated, opposing counsel discovers that as of Oct. 1, the software was actually at version 1.611. There may then be a question as to whether the best mode requirement was met.

To avoid source code version problems, the best mode should be met through higher-level diagrams: flow charts, block diagrams, or data flow diagrams.[384] The best mode should be described at such a level that the description is not affected by the typical day-to-day changes in source code. One commentator asserted that it is better to file a patent application for a computer program earlier on in the process when only relatively little is known about the best mode because, at that time, it is much easier to meet the best mode requirement.[385]

If the inventors believe that a specific piece of software works the "best" with their invention, that particular piece of software must be disclosed to meet the best mode requirement. An example given by Stephen A. Becker follows:

> As another example, if the inventor believes that a software implemented system benefits by a particular operating system, that operating system should be described in sufficient detail to enable a person skilled in the art to acquire the operating system, it if is commercially available, or to acquire or produce an equivalent operating system without need for undue experimentation.[386]

If particular subject matter is not claimed in the patent application, it is not necessary to disclose the best mode for this unclaimed subject matter.[387] Again, an example given by Stephen A. Becker is illustrative of this point:

> [A]ssume that the invention is a local area network system having a series of processors distributed along

> a common line and in communication with each other in accordance with a set of network priority rules. To satisfy the best mode requirement, the application will disclose the rules underlying network priority as well as the preferred manner of implementing them. However, the best mode requirement will not compel disclosure of specific data format unless the format is claimed in the application or is necessary for efficient implementation of the network rules.[388]

***281 1. Textual Description of Software Can Satisfy Best Mode**

Fonar Corp. v. General Electric Co.[389] involved the issue of a software patent meeting the best mode requirement. One patent at issue in the *Fonar* case was U.S. Patent Number 4,871,966, entitled "Apparatus and method for multiple angle oblique magnetic resonance imaging."[390] "The '966 patent concerns a technique for ***282** using a magnetic resonance imaging (MRI) machine in order to obtain multiple image slices of a patient's body at different angles in a single scan, referred to as multi-angle oblique (MAO) imaging."[391]

General Electric argued that the '966 patent did not satisfy the best mode requirement because (1) the patent failed to disclose two software routines[392] which the inventors testified were the best means they knew of to accomplish MAO imaging, (2) the patent failed to sufficiently disclosed a gradient multiplier board (GMB), and (3) the patent failed to identify a new integrated circuit "chip" for implementing certain functions of the hardware.[393]

At trial, a jury found that the '966 patent was not invalid.[394] The trial court then denied GE's motion for judgment as a matter of law.[395] GE appealed the denial of their motion for judgment as a matter of law to the Federal Circuit.[396] GE had the burden of showing that the jury's findings were not supported by substantial evidence or that the legal conclusion(s) implied from the jury's verdict cannot in law be supported by those findings.[397]

The Federal Circuit found that the jury's finding that the '966 patent satisfied the best mode requirement was supported by substantial evidence.[398] The court stated the general rule that "where software constitutes part of a best mode of carrying out an invention, description of such a best mode is satisfied by a disclosure of the functions of the software."[399] The court also agreed with witnesses of Fonar that "providing the functions of the software was more important than providing the computer code."[400] Judge Lourie stated that, once a software's functions have been disclosed, "normally, writing code for such software is within the skill of the art, not requiring undue experimentation."[401] The court found that flow charts and/or source ***283** code listings are not a requirement for adequately disclosing the functions of software.[402]

So, what exactly was disclosed in the '966 patent? The patent included no flow diagrams, source code listings, data flow diagrams, structure charts, or thread diagrams.[403] The terms "software," "computer program," "routine," "module," "procedure," and "function"[404] are not found in the entire patent. The specification does include a mathematical description of how to obtain, in the course of a single scan, image data for several differently oriented planes in an object, but the discussion is entirely mathematical and theoretical. The specification gives several equations for certain signals. Of course, anyone skilled in the art could write a computer program to calculate an output for these equations given the inputs. However, the specification discloses only limited practical aspects of implementing the invention (without flow diagrams, source code listings, data flow diagrams, structure charts, or thread diagrams).

The Federal Circuit also found that the finding "that the GMB was sufficiently disclosed to satisfy the best mode requirement was also supported by substantial evidence."[405] At trial, one of the inventors testified that the patent provided "a description of the functions required for one skilled in the art to build a GMB that will work with a general MRI system."[406] A witness of Fonar testified that the patent provided a description of the GMB within the dotted line in Figure 7 of the patent, reproduced below.[407]

TABULAR OR GRAPHIC MATERIAL SET FORTH AT THIS POINT IS NOT DISPLAYABLE

***284** The court also found substantial evidence supporting the jury's finding that the patent's functional description of a new chip to achieve certain functions was sufficient to satisfy the best mode requirement.[408] The '966 patent disclosed the functions of that chip in Figure 7 and provided a textual description of its functions. "Because adequate disclosure of the functions of the 'chip' was in the specification, failure to specifically identify a particular manufacturer's 'chip' was not fatal to satisfaction of the best mode requirement."[409]

The last two issues (the GMB and the new chip) both relied on Figure 7 of the ‘966 patent, shown above. The rest of the figures included waveforms, graphs, tabular data, a display, and a medical application example. The patent includes a description for each block found in Figure 7 and a description of the interfaces between the blocks. The interfaces were described at the level shown in Figure 7 (i.e., they were high-level interface descriptions).

Robotic Vision Systems affirmed the general rule of *Fonar*.[410] In *Robotic Vision Systems*, U.S. Patent 5,463,227 was at issue. “The ‘227 patent discloses a method of using a three-dimensional sensor in order to scan and inspect the leads of ***285** integrated circuit chips.”[411] The ‘277 patent improved the scanning time by reducing the number of changes in the direction of the scanner when compared with the prior art method.[412] The only claim in question at the Federal Circuit was claim 1,[413] which recited the row and column scanning feature.[414]

On a motion for summary judgment, View argued that the patent was invalid on the grounds of failure to disclose the best mode of carrying out the invention.[415] The trial court granted View’s motion for summary judgment, “concluding that, at the time the application for patent was filed, the inventors knew that using software was the only mode of practicing the invention, that it was undisclosed, and that the patent was accordingly invalid.”[416]

From the opinion, it seems that a problem may have been that View may have simply made a conclusory argument against the best mode requirement being met, without giving reasoning for such a conclusion. For example, the court pointed out that a witness averred that “A person of ordinary skill in the art to which the ‘227 patent pertains would know and understand that software is needed to perform the patented method. The details of such software would also be within the skill of a person of ordinary skill in the art to which the ‘227 patent pertains.”[417] After this statement, Judge Lourie noted that “View has not provided a basis for concluding that [these] assertions are not correct. Thus, one cannot conclude that a person skilled in the art would not have known that software was the best mode of carrying out the invention and how to implement it.”[418]

The court then addressed the lack of source code in the patent application. Judge Lourie recited the general rule from *Fonar* “that when disclosure of software is required, it is *generally* sufficient if the functions of the software are disclosed, it *usually* being the case that creation of the specific source code is within the skill of ***286** the art.”[419] It is important to note the latitude in the words “generally” and “usually” used in stating the general rule. The Federal Circuit is leaving room so that, possibly in the next case involving best mode, a textual description of how something is done may not be enough to satisfy the best mode requirement. Again, View did not present any evidence to controvert the assertion that “one skilled in the art could generate the necessary software program to implement the disclosed functions.”[420] The Federal Circuit then concluded that the district court erred in granting View’s motion for summary judgment that the patent was invalid for failure to disclose the best mode.[421]

The *Fonar* and *Robotic Vision Systems* illustrate that it is not an absolute requirement that the author include source code, flow diagrams, and other similar software specific figures to meet the best mode requirement for software. *Fonar* and *Robotic Vision Systems do not* stand for the proposition that the patent doesn’t need flow diagrams (or the like) and in some cases source code to satisfy best mode. In other words, when software is involved in the best mode of practicing the invention, a textual description of how the computer program works and what it does should be included at a minimum. A prudent author should include, for software, not only a textual description of what the program does, but also a flow diagram, and a data flow diagram (or something similar).[422] The facts of these two cases have a serious impact on the power of the *Fonar* rule. *Fonar* involved a court finding whether there was substantial evidence to support the jury’s findings. *Robotic Vision Systems* was on a summary judgment standard, and the defendant failed to bring forth evidence to support its assertions that the best mode was not met. The use of the words “generally” and “usually” in reciting the rule in *Robotic Vision Systems* indicates that the court was setting the textual description only as a minimum requirement—meaning that a patent would need *at least* a textual description of the software to satisfy the best mode requirement.

When will a textual description only, without flow diagrams or the like, not satisfy best mode? Perhaps when an argument is made that best mode is not satisfied and some evidence is brought in to support the argument, or perhaps when a court decides a best mode issue and not a jury will we find that a simple textual description is not enough. For enablement and best mode, a patent author should ***287** always include at least one or more flow diagrams[423] for the software claimed, and probably several more software specific diagrams (data flow diagrams, structure charts, thread diagrams, or object diagrams). Most likely, only in exceptional circumstances would the author include source code.

VI. Recently Issued Software Patents

Focusing on principles alone may leave a practitioner, especially a fairly inexperienced practitioner, floating in the world of theory. To help bring a real-world perspective into this discussion of software patents, this Section presents a case study of a few software patents that have recently issued. The only criterion used to select patents was that the claims had to be clearly directed toward software. No attempt was made to evaluate the quality of the patents or claims analyzed. The patents analyzed in the study were assigned to the respective companies of Apple, IBM, Microsoft, and Novell.

Each patent has been analyzed for several statistics. A table will be shown for each assignee containing these statistics. The statistics include the total number of claims, divided into the number of independent claims and the number of dependent claims. Each independent claim was categorized as being an apparatus claim, a method claim, or a data structure type claim (similar to the claim reproduced above in *Lowry*). The tables list the ratio of the type of claim to the total number of independent claims. The row labeled "ind. app. claims/total inds." indicates the ratio of the total number of independent apparatus claims to the total number of independent claims. The row labeled "ind. method claims/total inds." indicates the ratio of the total number of independent method claims to the total number of independent claims. Finally, the row labeled "ind. data structure claim/total inds." indicates the ratio of the total number of independent data structure type claims to the total number of independent claims. For the apparatus claims, whether the patent employed means-plus-function language in any of its apparatus claims is indicated in the row labeled "means-plus-function language used?."

The length of different parts of each patent were also determined. The smallest independent claim was listed as the total number of words found in that claim, the longest independent claim was also listed, as well as the average number of words found in the independent claims in that patent. These numbers can be misleading. The breadth of a claim should not be assumed to be directly related to the length of the claim, although it usually is.

***288** Because the specification is very important to a patent, the statistics include the length of the background section, the length of the summary section, the length of the detailed description, and the number of figures included in the patent. These numbers provide no concrete rules for drafting the specification but may be useful for comparison purposes.

Of course, several very important statistics and other pieces of information are unknown and may have affected the statistics presented. For instance, the method of payment for the patent application to be filed and prosecuted is a factor that may affect the length of the specification and the number of claims. A tremendous difference would likely exist between statistics for a client paying a flat rate per patent and a client paying an hourly rate with a primary concern for the quality of the patent. Another factor that may affect the number of claims allowed is the tenacity of the Examiner considering the patent and his or her experience in the software field. Furthermore, the number of existing patents in the application area of the invention may affect the thoroughness with which the Examiner considered the patent.

Following the statistics for patents assigned to each assignee is a brief analysis of randomly selected claims.

A. Software Patents Assigned to Apple

The study included several patents that had been assigned to Apple Computer, Inc. These patents were U.S. Patent No. 5,671,446 with an issue date of Sep. 23, 1997 (legal representation listed on patent as Hickman, Beyer & Weaver, LLP); U.S. Patent No. 5,671,438 with an issue date of Sep. 23, 1997 (legal representation listed on patent as Hickman, Beyer & Weaver, LLP); U.S. Patent No. 5,670,986 with an issue date of Sep. 23, 1997 (legal representation listed on patent as Blakely, Sokoloff, Taylor & Zafman); U.S. Patent No. 5,669,005 with an issue date of Sep. 16, 1997 (legal representation listed on patent as Burns, Doane, Swecker & Mathis); U.S. Patent No. 5,669,000 with an issue date of Sep. 16, 1997 (legal representation listed on patent as Blakely, Sokoloff, Taylor & Zafman); U.S. Patent No. 5,666,552 with an issue date of Sep. 9, 1997 (legal representation listed on patent as Blakely, Sokoloff, Taylor & Zafman); and U.S. Patent No. 5,860,079 with an issue date of Jan. 12, 1999 (legal representation listed on patent as Burns, Doane, Swecker & Mathis).

Statistics for these patents are presented in the following table.

289** ***Table 1. Statistics for Patents Assigned to Apple.

	5671446	5671438	5670986	5669005	5669000	5666552	5860079
Total number of claims	43	26	13	12	7	23	5
Number of independent claims	6	4	3	4	2	6	3
Number of dependent claims	37	22	10	8	5	17	12
Ind. app. claims/total inds.	2/6	1/4	1/3	1/4	0/2	3/6	2/3
Means-plus-function language used?	Y	Y	N	Y	-	Y	Y
Ind. method claims/total inds.	3/6	3/4	2/3	3/4	0/2	3/6	1/3
Ind. data structure claim[424]/total inds.	1/6	0/4	0/3	0/4	2/2	0/6	0/3
Smallest independent claim[425]	99	77	153	151	100	210	147
Longest independent claim	247	132	210	201	157	325	202
Ave. # words in independent claims[426]	154	100	172	179	129	270	167
Length of background section (# wrds)	1123	502	771	868	673	1273	952
Length of the summary[427] (# words)	1015	538	499	672	872	645	455
Length of detailed description (# wrds)	7901	5568	3879	9364	16448	4877	3461
Number of figures	14	14	10	32	11	13	10

From the above patents assigned to Apple, one apparatus claim, one method claim, and one data structure claim will be analyzed. The claims listed below were chosen randomly. An example of an apparatus claim is found in U. S. Patent No. 5,666,552, *Method and apparatus for the manipulation of text on a computer display screen*, claim 6:
6. In an interactive computer-controlled display system[A1] having a processor[A2], a memory means[A3], a display device[A4] coupled to said processor for visibly displaying text, ***290** a cursor control device[A5] coupled to said processor, a device for directly manipulating a portion of text displayed on said display device, said device comprising:
means for selecting a block of text[A6] from a set of available text, at least a portion of said available text displayed on said display device[A7];

means for dragging said selected block[A8] of text from a previous position to a new position relative to said available text, at least a portion of said selected block of text being visible as said selected block of text is dragged, said movement of said selected block of text being responsive to movement of said cursor control device[A9];

means for displaying an insertion marker on said display device[A10], said insertion marker specifying a position in said available text where said selected block of text is inserted by a means for inserting; and

said means for inserting, inserting said selected block of text into said available text, said selected block of text inserted at said new position, said means for inserting reformatting at least a part of said available text such that the at least a part of the available text is unobscured by the insertion of the selected block of text and to provide continuity between said available text and said selected block of text.[428]

In the ‘552 patent, claim 6, the author set up the hardware environment in the claim preamble. Elements **A1** through **A5** are hardware elements needed for the claimed invention. Element **A6**, a “block of text” is a real-world element, especially to anyone that has used a word processor. In addition, most computer users have used a windows environment and know what “dragging” a block of text means, as used in element **A8**. Elements **A7** and **A10** tie back into the claim preamble’s recitation of display device. Similarly, element **A9** also ties back into the claim preamble by relating back to the “cursor control device.”

An example of a method claim is found in U. S. Patent No. 5,670,986, *Graphics system for displaying images in gray-scale*,

claim 1:
1. In a computer graphics system[B1], a method for displaying a color image[B2] at lower color resolution, the color image comprising a plurality of pixels[B3] each having one of a plurality of colors, the method comprising the steps of:
defining a three-dimensional color space as a plurality of discrete luminance levels; generating a luminance lookup table[B4], said table comprising a plurality of index values[B5] each representing one of said plurality of discrete luminance levels; for a pixel[B6] in the color image[B7], mapping a pixel color of the pixel to a selected one of said index values that out of said plurality of index values represents a first luminance level that is nearer to a luminance level associated with said pixel color than a second luminance level represented by any other of said plurality of index values;

and rendering the color image[B8] using the selected one of said index values in the luminance lookup table.[429]

***291** Element **B1** in the claim preamble sets up the physical environment in which the invention will operate. However, nothing else in the claim seems to directly relate back to this physical environment. **B1** may simply be a field of use label included in the claim preamble. Arguably elements **B2** and **B3** are parts of this system, but not by recitation in the claim. **B2** and **B3**, the color image and pixels, may be more difficult for most people to recognize as physical objects. **B4** and **B5**, the table and index values, also attempt to relate to the real world but are abstract concepts that may be unfamiliar to most computer users. **B6, B7**, and **B8** all relate to the claim preamble's recitation of color image and pixels. Overall, this claim relies on its ties to the color image and pixels to recite a practical application. The Examiner was apparently satisfied with this approach because the claim was allowed. The claim relates only loosely to real-world elements, but it recites patentable subject matter according to this examiner.

An example of a data structure type claim is found in U. S. Patent No. 5,671,446, *Method and apparatus for atomically accessing a queue in a memory structure where LIFO is converted to FIFO*, claim 39:
39. A queue memory structure accessible by a plurality of enqueuers[C1] and dequeuers[C2] implemented in a computer system[C3], the queue memory structure comprising:
a read/write memory device[C4] coupled to said computer system[C5] for providing said queue memory structure;

a LIFO structure implemented in said read/write memory[C6] for storing data elements input to said queue memory structure by an enqueuer, wherein said LIFO is atomically accessed by said enqueuer such that said enqueuer cannot be interrupted by other enqueuers accessing said queue memory structure when said enqueuer is storing said data elements;

a FIFO list implemented in said read/write memory[C7] for storing data elements to be output from said queue memory structure by a dequeuer, wherein said LIFO is atomically accessed by said enqueuer such that said enqueuer cannot be interrupted by other enqueuers accessing said queue memory structure when said enqueuer is storing said data elements, and wherein said LIFO is reversed to create said FIFO by said dequeuer when said FIFO is empty; and

a lock flag implemented in a memory device[C8] coupled to said computer system[C9] that can be set by a dequeuer accessing said queue memory structure, said lock indicating that said queue memory structure is currently being accessed and may not be accessed by different dequeuers while said lock is set,

whereby said data elements in said queue memory structure are stored and retrieved in a first-in-first-out order without altering the level of interrupts used on said computer system[C10] and without risk of deadlock.[430]

***292** The preamble sets up the claim as being implemented in a computer system by **C3** and continues to relate back to the computer system with **C5, C9**, and **C10**. The claim also ties a memory device into the computer system using **C4**. In addition, the preamble recites software components, the enqueuers and dequeuers, in **C1** and **C2**, that interact with the claimed structure. Note that the LIFO structure, the FIFO list, and the lock flag are not claimed in the abstract, but are claimed as being implemented in a memory device as shown by elements **C6, C7**, and **C8**. This claim follows the Guidelines, which state that data structures encoded on computer-readable medium are statutory.[431]

B. Software Patents Assigned to IBM

Several patents assigned to IBM were U.S. Patent No. 5,652,864 with an issue date of Jul. 29, 1997 (legal representation not listed); U.S. Patent No. 5,594,910 with an issue date of Jan. 14, 1997 (legal representation listed on patent as Paul C. Scifo); U.S. Patent No. 5,586,296 with an issue date of Dec. 17, 1996 (legal representation listed on patent as Ronald L. Drumheller); U.S. Patent No. 5,822,763 with an issue date of Oct. 13, 1998 (legal representation listed on patent as Tassinari; Robert P. Cameron; Douglas W. Dougherty; Anne Vachon); U.S. Patent No. 5,809,471 with an issue date of Sep. 15, 1998 (legal representation listed on patent as Dougherty; Anne Vachon; Daniel P. Morris); and U.S. Patent No. 5,808,620 with an issue date of Sep. 15, 1998 (legal representation listed on patent as Perman & Green).

Statistics for these patents are presented in the following table.

293** ***Table 2. Statistics for Patents Assigned to IBM.

	5652864	**5594910**	**5586296**	**5822763**	**5809471**	**5808620**
Total number of claims	18	12	7	16	37	15
Number of independents	6	2	2	3	7	3
Number of dependent claims	12	10	5	13	30	12
Ind. app. claims/total inds.	1/6	1/2	1/2	1/3	4/7	1/3
Means-plus-function language used?	Y	Y	Y	Y	Y	Y
Ind. method claims/total inds.	4/6	1/2	1/2	2/3	3/7	2/3
Ind. data structure claim[432]/total inds.	1/6	0/2	0/2	0/3	0/7	0/3
Smallest independent claim[433]	200	317	159	169	54	218
Longest independent claim	239	376	194	204	181	285
Ave. # words in independent claims[434]	218	347	177	186	108	244
Length of background section (# wrds)	611	578	403	392	287	872
Length of the summary[435] (# words)	345	678	470	120	1111	305
Length of detailed description (# wrds)	5838	29750	2018	4806	2237	5696
Number of figures	15	16	8	5	1	26

An example of an apparatus claim is found in U. S. Patent No. 5,594,910, *Interactive computer network and method of operation*, claim 7:
7. A computer network[D1] for providing a multiplicity of users access to a multiplicity of applications[D2], the applications each including data, the apparatus comprising:
***294** a. one or more host computers[D3], each including a data store[D4] containing data used in creating applications[D5];

b. a plurality of concentrator computers[D6] connected[D7] in groups of one or more to each of the host computers, each of the concentrator computers including a data store[D8] containing data used in creating applications[D9];

c. a plurality of reception system computers[D10] at which respective users can request applications[D11], the reception system computers being connected [D12] in groups of one or more to each of the concentrator computers, the reception system computers each including a data store[D13] containing data used in creating applications[D14];

d. the respective data stores[D15] of the host computers, the concentrator computers and the reception system computers being responsive to a control attribute ascribed to the application data for dictation, at least in part, storage of the application data at the respective data stores; and

e. data distribution means[D16] for distributing data in the network[D17] such that data required for an application requested at a respective reception system[D18] may be collected from the data store of the respective reception system and the data stores of the host computer[D19] and concentrator computer[D20] to which the respective reception system is connected and wherein the data distribution means includes means provided at the respective reception system, for determining whether the requested application can be constituted from data stored at the respective reception system, and to the extent it is determined that required data is not stored at the respective reception system, requesting the required data from the network[D21], the data distribution means further includes means for maintaining data at the data stores of the network dependent upon the likelihood an application associated with the data will be requested so that data required for an application likely to be requested is likely to be located at the respective reception system and data required for applications least likely to be requested is not likely to be located at the respective reception computer and wherein the means for maintaining data at the network data stores retains data dependent in part upon preferences of the respective users of the requesting reception systems.[436]

The claim relates to physical things rather than abstract ideas or notions. The context is a computer network as recited in the preamble at **D1**. Elements **D17** and **D21** also refer to the network. Different computers at **D3, D6**, and **D10** are claimed as part of this system. These different computers have data stores (also real-world elements) at **D4, D8**, and **D13**, which are further described in the paragraph starting with element **D15.** At **D5, D9**, and **D14** are claimed specific data contained in these data stores, which also tie back into the claim preamble at **D2** where it recites applications. **D11** also ties back into the applications that users can request. The different computers are connected and interrelated by **D7** and **D12.** The means clause at **D16** is related to the network at **D17**, and further related to the different computers claimed at **D18, D19**, and **D20.**

***295** An example of a method claim is found in U. S. Patent No. 5,586,296, *Cache control system and method for selectively performing a non-cache access for instruction data depending on memory line access frequency*, claim 6:
6. A method of operating a computer system comprising a processor[E1], a main memory[E2] and a cache memory[E3] in which memory locations[E4] are arranged in memory lines[E5], each line having an index associated therewith, the method comprising, maintaining, for each of a number of the memory lines[E6], a stored frequency value indicative of the number of accesses made thereto;

maintaining, in an instruction store[E7], the index of the last memory line [E8] accessed by each of a number of instructions;

and, in the event of a cache miss, testing the stored frequency value associated with the last memory line[E9] accessed by the current instruction,

and if the number of times that the last memory line[E10] has been accessed has passed a threshold value, fetching the memory line[E11] into the cache[E12] and executing a cached memory access,

and if the number of times that the last memory line[E13] has been accessed has not passed the threshold value, executing a non-cached memory access.[437]

This method claim's preamble sets up the physical environment in which this method will be accomplished by reciting different pieces of hardware at **E1, E2**, and **E3.** It further defines **E3** at **E4** and **E5.** The memory lines first recited at **E5** are continually referred to throughout the claim as shown by **E6, E8, E9, E10, E11**, and **E13.** This claim continually reminds the reader of the practical application of this invention. **E12** apparently refers back to the cache first introduced in the claim preamble at **E3.** An instruction store, a real-world element, is introduced at **E7**.

The following claim is a hybrid of method, apparatus, and data structure claims. With its use of means-plus-function language, it is unclear whether this claim protects an apparatus, a method, or a data structure. This claim is found in U. S. Patent No. 5,652,864, *Concurrent storage allocations or returns without need to lock free storage chain*, claim 18:
18. A computer program product[F1] for allocating storage blocks, said program product comprising:

a computer readable medium[F2];
first program instruction means[F3] for instructing a processor to chain together available storage blocks and store in each block

a length of the respective block and a pointer to a next block in the chain;

second program instruction means[F4] for instructing a processor to search the chain to identify one of the available blocks with a length large enough to ***296** satisfy one storage allocation request while permitting allocation of another block on the chain pursuant to another storage allocation request;

third program instruction means[F5] for instructing a processor to record the length of said one available block after said one available block is identified; and

fourth program instruction means[F6] for instructing a processor, while preventing access to said one available block pursuant to said other request, to compare a length of said one available block currently stored in said one available block to the recorded length to determine if the length of said one available block was changed since the length was recorded, and if not, reduce the length of said one available block by an amount equal or greater than that of said one request such that after the preventing step another storage allocation request cannot be satisfied from the length by which said one available block was reduced; and

wherein each of said program instruction means is recorded on said medium.[F7][438]

This claim recites a computer program product at **F1**. This product comprises a computer readable medium at **F2**, as well as four instruction means at **F3, F4, F5**, and **F6**. Finally, in the last clause, the claim places each of the instructions means onto the computer readable medium at **F7**.

C. Software Patents Assigned to Microsoft

Several patents assigned to Microsoft were U.S. Patent No. 5,673,394 with an issue date of Sep. 30, 1997 (legal representation on patent listed as Seed and Berry LLP); U.S. Patent No. 5,664,191, with an issue date of Sep. 2, 1997 (legal representation listed on patent as Seed and Berry LLP); U.S. Patent No. 5,664,133 with an issue date of Sep. 2, 1997 (legal representation listed on patent as Seed and Berry LLP); U.S. Patent No.5,659,685 with an issue date of Aug. 19, 1997 (legal representation listed on patent as Leydig, Voit & Mayer, Ltd.); U.S. Patent No.5,659,336 with an issue date of Aug. 19, 1997 (legal representation listed on patent as Klarquist Sparkman Campbell Leigh & Whinston, LLP); U.S. Patent No. 5,655,148 with an issue date of Aug. 5, 1997 (legal representation listed on patent as Jones & Askew); and U.S. Patent No. 5,864,669 with an issue date of Jan. 26, 1999 (legal representation listed on patent as Workman, Nydegger & Seeley).

Statistics for these patents are presented in the following table.

297** ***Table 3. Statistics for Patents Assigned to Microsoft.

	5673394	**5664191**	**5664133**	**5659685**	**5659336**	**5655148**	**5864669**
Total number of claims	11	13	38	12	20	5	6
Number of independent claims	5	8	8	2	3	1	2
Number of dependent claims	6	5	30	10	17	4	4
Ind. app. claims/total inds.	0/5	2/8	3/8	1/2	1/3	0/1	1/2
Means-plus-function language used?	-	Y	Y	Y	N	-	Y
Ind. method claims/total inds.	5/5	6/8	5/8	1/2	2/3	1/1	1/2
Ind. data structure claim[439]/total inds.	0/5	0/8	0/8	0/2	0/3	0/1	0/2
Smallest independent claim[440]	166	97	94	152	139	243	193

Longest independent claim	304	143	150	168	246	243	227
Ave. # words in independent claims[441]	244	124	134	160	187	243	210
Length of background section (# wrds)	669	689	703	900	1200	1094	812
Length of the summary[442] (# words)	243	481	565	367	444	4226	807
Length of detailed description (# wrds)	7327	5720	4162	7030	2478	31405	1051
Number of figures	22	10	12	6	8	25	3

An example of an apparatus claim is found in U. S. Patent No. 5,659,685, *Method and apparatus for maintaining network communications on a computer capable of connecting to a WAN and LAN*, claim 12:
***298** 12. A networked computer[G1] for dynamically supporting connections to a plurality of networks including at least one local area network (LAN) connection[G2] and a wide area network (WAN) connection[G3], the networked computer comprising:
a WAN adaptor connection controller[G4] for detecting an up WAN link to a remote networked computer;

means for[G5] determining an initiator of the up WAN link; and

means for[G6] configuring a set of network components of the networked computer, the means for configuring comprising:

means for[G7] disconnecting existing logical network connections between the networked computer and all physically coupled networks except a network associated with the up WAN link in response to a determination that the networked computer initiated the up WAN link; and

means for[G8] enabling routing of packets, by the networked computer, between at least the WAN link and a second connected network in response to a determination that the remote networked computer initiated the up WAN link.[443]

This apparatus claim used means-plus-function language almost exclusively in its claim elements. The preamble set up the environment as a networked computer at **G1** with real network connections at **G2** and **G3**. Element **G4** is a real-world element, a controller. The rest of the explicity claimed elements are all means-plus-function elements, as shown by **G5, G6, G7**, and **G8**.

An example of a method claim is found in U. S. Patent No. 5,664,133, *Context sensitive menu system/menu behavior*, claim 1:
1. In a computer system having a central processing unit (CPU), a graphical user interface including a display and a user interface selection device communicatively coupled to the CPU, a method for providing, and selecting from, a menu for a selected computer resource, said method comprising the steps of:
generating a set of menu selections for the selected computer resource in response to receiving, by the CPU, a context menu generation signal from the user interface selection device, the generating step comprising the steps of:

retrieving a menu selection relating to a class of objects to which the selected computer resource belongs; and

retrieving a menu selection associated with a container in which the selected computer resource resides; and

displaying upon the display the set of menu selections in a menu positioned in the proximity of a graphical representation of the selected computer resource.[444]

An example of an interesting method claim that suggests a data structure claim by its preamble was found in U. S. Patent No. 5,664,191, *Method and system for* ***299** *improving the locality of memory references during execution of a computer program*, claim 10:
10. A computer-readable medium[I1] whose contents cause a computer system to reduce the size of code that includes live code

segments and dead code segments <u>by performing the steps of</u>[I2]:
locating a conditional branch from a live code segment to a dead code segment;

replacing the located conditional branch with a conditional branch to the live code segment;

storing the address of the conditional branch to the live code segment and the address of the dead code segment in a storage data structure outside of the live code segment; and

providing a monitoring means for detecting execution of the conditional branch to the live code segment during execution of the live code, and, upon detection, searching the storage data structure for the address of the dead code segment corresponding to the address of the conditional branch, and causing execution of the dead code segment.[445]

The above claim is similar in approach to the claim recited above from U. S. Patent No. 5,652,864. This claim also uses a computer readable medium at **I1**, but the claim does not directly relate structures stored on that medium. The '864 claim used means-plus-function language to describe what was stored on the medium, while this claim uses method steps to describe what is on the storage medium as shown by **I2**. One reason for the common approaches may be that the authors felt that directly reciting the data structures on the medium unnecessarily limits the claims, and that reciting functional language is broader.

D. Software Patents Assigned to Novell

Several patents assigned to Novell were U.S. Patent No. 5,666,532 with an issue date of Sep. 9, 1997 (legal representation on patent listed as Computer Law); U.S. Patent No. 5,608,903 with an issue date of Mar. 4, 1997 (legal representation on patent listed as Dinsmore & Shohl); U.S. Patent No. 5,596,574 with an issue date of Jan. 21, 1997 (legal representation on patent listed as Cesari and McKenna); U.S. Patent No. 5,553,139 with an issue date of Sep. 3, 1996 (legal representation on patent listed as Snell & Wilmer); U.S. Patent No. 5,428,738 with an issue date of Jun. 27, 1995 (legal representation on patent listed as Weil, Gotshal & Manges); U.S. Patent No. 5,856,974 with an issue date of Jan. 5, 1999 (legal representation on patent listed as Cesari and McKenna); and ***300** U.S. Patent No. 5,808,751 with an issue date of Sep. 15, 1998 (legal representation on patent listed as Schweitzer Cornman Gross & Bondell).

Statistics for these patents are presented in the following table.

Table 4. Statistics for Patents Assigned to Novell.

	5666532	**5608903**	**5596574**	**5553139**	**5428738**	**5856974**	**5808751**
Total number of claims	52	21	19	4	10	14	6
Number of independent claims	4	4	5	1	2	7	1
Number of dependent claims	48	17	14	3	8	7	5
Ind. app. claims/total inds.	2/4	0/4	2/5	0/1	1/2	3/7	1/1
Means-plus-function language used?	Y	-	Y	-	Y	Y	Y
Ind. method claims/total inds.	2/4	4/4	3/5	1/1	1/2	4/7	0/1
Ind. data structure claim[446]/total inds.	0/4	0/4	0/5	0/1	0/2	0/7	0/1
Smallest independent claim[447]	266	236	98	91	175	235	218
Longest independent claim	373	421	130	91	204	307	218
Ave. # words in independent claims	322	306	116	91	190	264	218

Length of background section (# wrds	875	1071	1253	472	197	2041	494
Length of the summary (# words)	429	57	686	376	103	778	306
Length of detailed description (# wrds	16352	4166	2421	3132	3473	3936	1771
Number of figures	20	8	8	8	11	8	4

An example of an apparatus claim is found in U. S. Patent No. 5,596,574, *Method and apparatus for synchronizing data transmission with on-demand links of a network*, claim 1:
1. Apparatus for synchronizing delivery of a data packet generated by a source node and transmitted over an on-demand link of a computer network to a destination node in a manner that efficiently utilizes that link, the apparatus comprising:
a router[J1] coupled to the on-demand link, the router configured[J2] to activate the link to establish a connection to another node of the network for delivery of the packet to the destination node; and

***301** means for instructing the router[J3] whether to immediately activate the link to establish the connection, the instructing means comprising control information[J4] stored in the data packet.[448]

The claimed device comprises a data packet router at **J1**, configured a certain way at **J2**, and a means for instructing the router at **J3**, comprising control information in the data packet at **J4**. The real-world practical application is the delivery of the data packet from one computer node to another.

An example of a method claim is found in U. S. Patent No. 5,666,532, *Computer method and apparatus for asynchronous ordered operations*, claim 29:
29. A method in a computer system having a first unit and second unit for files, having a file management subsystem for controlling operations for files, said file management subsystem specifying operations for files in response to new requests where a sequence of requests for the operations is represented by the requests R1, R2, …, Rr and where the requests for the operations in said sequence have order dependencies D1, D2, …, Dd where r and d are integers, said order dependencies constraining the order for carrying out the operations, said computer system including an ordering subsystem for controlling the order of operations including, said method comprising:
storing a plurality of entries in an ordering store, each of said entries containing an operation type identifying one of said operations for files, at least one of said entries at some time also containing a link which links said entry to another of said entries, said link specifying an order for carrying out said operations in said linked entries, said entries and said links defining a partially ordered acyclic graph,

adding entries to the ordering store by processing said new requests to identify one or more common operations CO1, CO2, …, COco, each of the common operations identifying an operation requested by one or more of the requests R1, R2, …, Rr, where said common operations have common order dependencies CD1, CD2, …, CDcd that preserve the order dependencies D1, D2, …, Dd between the operations in the requests, and where co and cd are integers, and

executing said one or more common operations CO1, CO2, …, Coco responsive to the entries in the ordering store.[449]

VII. Summary of Principles to Follow in Drafting Software Patent Applications

No fourth category of subject matter is excluded from Section 101 under the label of a mathematical algorithm.[450] Disembodied mathematical concepts are nonstatutory because they represent nothing more than “laws of nature,” “natural ***302** phenomenon,” or “abstract ideas.”[451] After *State Street*, all that is required to overcome this hurdle is to have the claimed invention be embodied in a computer program and show that it is “useful.”[452] The Federal Circuit in *State Street* also clarified that there is no longer a “business method exception” to statutory subject matter.[453] Patent claims on methods of doing business now clearly stand on equal footing with other types of claims in terms of statutory subject matter.

Claims should have a real-world, practical application.[454] The more the claim transforms or reduces a physical object to a different state or thing, the more likely it will be found to comply with the statute.[455] A mere field of use label in the claim preamble may not be sufficient to make the claim statutory. Rather than claiming something that merely calculates a number, tie claim elements to the environment and field of use recited in the preamble.[456]

A claim will be found to be statutory if it results in a physical transformation outside the computer.[457] The Guidelines refer to specific categories (safe harbors) within which a claimed process will be found to have resulted in a physical transformation outside the computer.[458] The safe harbors are "independent physical acts (post-computer process activity)" and "manipulation of data representing physical objects or activities (precomputer process activity)."[459]

Draft the claims for a patent before drafting the detailed description.[460] Drafting the claims before the specification distinctly points out the invention to the patent author thereby telling him or her exactly what must be supported in the specification.[461]

***303** The claim set should include several independent claims, of varying scope and disparate approaches.[462] Drafting claims using different approaches means that a patent application for a piece of software typically will have both apparatus and method claims.[463] A data structure claim should be included to capture manufactures of software distributed on CD-ROM as direct infringers.[464] When claiming data structures, emphasize the physical organization that is required by the data structures; do not claim the data structures in the abstract where any conceivable physical organization would read on the claim.[465]

Apply Section 112 ¶ 6 to means-plus-function claim elements when determining patentability.[466] Because claims with and without means-plus-function claim limitations have advantages, both types of claims should be included with the patent application.[467]

Always draft a specification containing a full disclosure of the invention, the environment in which it operates, and how the invention works in that specific environment.[468] Ensure that the specification explains the steps involved in the method claims, and the blocks included in the apparatus claims.[469] A deficient specification may indicate to a court that the author didn't really have a real-world problem in mind, or that the author didn't really understand how this abstract idea may be used in a practical application.[470] In addition, include flow charts with the patent application.[471]

Source code is usually not needed for enablement.[472] A skilled programmer can write routines to accomplish specific tasks; he or she does not need to be given the source code to do it.[473] For software, it is probably a good idea to include at least ***304** one figure directed towards the hardware environment in which the software will run.[474]

Relying on source code for best mode is not a good idea.[475] Rather, the best mode should be met through higher-level diagrams: flow charts, block diagrams, data flow diagrams, etc.[476] The best mode should be described at such a level that the description is not affected by typical day-to-day changes in source code.[477]

VIII. Conclusion

From recent case law and administrative guidance, patent authors can glean many principles that will enable them to claim software inventions so that the requirements for statutory subject matter, enablement, and best mode are met. Patenting computer-related inventions may have had a rough start, but progress is being made. As time progresses, patent attorneys and agents will gain experience in the software world, and so will the Examiners. The more experience we gain with the technology, the easier it will be to apply the statutory requirements to patent applications for computer-related inventions.

Footnotes

a1 B.S.E.E. *summa cum laude*, C.S. minor, Utah State University, 1993; Software Engineer, Motorola, Inc., 1993-1995; J.D. University of Utah College of Law, 1998; Registered Patent Attorney with the Intellectual Property and Technology Law Group of Parsons Behle & Latimer, Salt Lake City, Utah. This article represents the opinions of the author and does not necessarily represent those of his firm or any of its clients.

1 *See* Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1025-26 (1990) (arguing that "there is a basis in patent law for denying

patents to computer program algorithms and to a number of other computer program-related innovations" and questioning whether the patent system should be used to "protect program innovations that lie in the representation, organization, manipulation, and display of information").

2 *See, e.g.*, GREGORY A. STOBBS, SOFTWARE PATENTS §11.1 at 348 (1995 & Supp. 1998).

3 Brian R. Yoshida, *Claiming Electronic and Software Technologies: The Effect of the Federal Circuit Decisions in Alappat, Warmerdam, and Lowry on the Claiming of Mathematical Algorithms and Data Structures*, 45 BUFFALO L. REV. 457, 458 (1997) (stating that "new and useful computer program-related inventions and algorithms, including mathematical algorithms, should constitute subject matter eligible for patent protection as processes or machines"); *see* Maximilian R. Peterson, Note, *Now You See It, Now You Don't: Was It a Patentable Machine or an Unpatentable "Algorithm"? On Principle and Expediency in Current Patent Law Doctrines Relating to Computer-Implemented Inventions*, 64 GEO. WASH. L. REV. 90, 92 (1995) (asserting that "current judicial doctrines unfairly deny patent protection to the innovative concepts embodied in many computer-implemented inventions").

4 *See, e.g.*, Thomas P. Burke, Note, *Software Patent Protection: Debugging the Current System*, 69 NOTRE DAME L. REV. 1115 (1994); Jonathan E. Retsky, *Computer Software Protection in 1996: a Practitioner's Nightmare*, 29 J. MARSHALL L. REV. 853, 854 (1996) (proposing an "alternate and separate scheme of computer software protection"); John A. Burtis, Comment, *Towards a Rational Jurisprudence of Computer-Related Patentability in Light of In re Alappat*, 79 MINN. L. REV. 1129, 1130 (1995) (proposing a "new model for determining patentable subject matter"); Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308 (1994) (suggesting a *sui generis* system to protect computer software).

5 Michael A. Dryja, *Looking to the Changing Nature of Software for Clues to its Protection*, 3 U. BALT. INTELL. PROP. L.J. 109, 131 & n.126 (1995) (citing David Bender & Anthony R. Barkume, *Patents for Software-Related Inventions*, 5 SOFTWARE L.J. 279, 280-81 n.2 (1992) (advocating patent protection of software)).

6 *See, e.g.*, Judith A. Szepesi, Comment, *Maximizing Protection For Computer Software*, 12 SANTA CLARA COMPUTER & HIGH TECH. L.J. 173 (1996) (discussing patent, copyright, and trade secret protection for computer software and their strengths and weaknesses).

7 For an in-depth analysis of case law in this area, and its historical development, *see, e.g.*, 1 DONALD S. CHISUM, CHISUM ON PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY AND INFRINGEMENT § 1.03[6] (1997); GREGORY A. STOBBS, SOFTWARE PATENTS, ch. 9 (1995).

8 409 U.S. 63, 175 U.S.P.Q. (BNA) 673 (1972).

9 437 U.S. 584, 198 U.S.P.Q. (BNA) 193 (1978).

10 450 U.S. 175, 209 U.S.P.Q. (BNA) 1 (1981).

11 *Gottschalk*, 409 U.S. at 64, 175 U.S.P.Q. at 674.

12 Claim 8 is analyzed in detail in the text of the article. Claim 13 is not discussed, but reads as follows:
13. A data processing method for converting binary coded decimal number representations into binary number representations comprising the steps of
(1) testing each binary digit position i beginning with the least significant binary digit position, of the most significant decimal digit representation for a binary '0' or a binary '1';
(2) if a binary '0' is detected, repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation;
(3) if a binary '1' is detected, adding a binary '1' at the (i + 1)th and (i + 3)th least significant binary digit positions of the next

lesser significant decimal digit representation, and repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation;
(4) upon exhausting the binary digit positions of said most significant decimal digit representation, repeating steps (1) through (3) for the next lesser significant decimal digit representation as modified by the previous execution of steps (1) through (3); and
(5) repeating steps (1) through (4) until the second least significant decimal digit representation has been so processed.
Id. at 74, 175 U.S.P.Q. at 677.

13 *Id.* at 64, 175 U.S.P.Q. at 674.

14 *Id.* at 73-74, 175 U.S.P.Q. at 677.

15 *Id.* at 71-72, 175 U.S.P.Q. at 676.

16 *Id.* at 67, 175 U.S.P.Q. at 675 (citing Rubber-Tip Pencil Co. v. Howard, 87 U.S. 498, 507 (1874)).

17 *Id.*

18 *Id.* (citing Funk Bros. Seed Co. v. Kalo Co., 333 U.S. 127, 130, 76 U.S.P.Q (BNA) 280, 281 (1948)).

19 *Id.* at 68, 175 U.S.P.Q. at 675.

20 *Id.* at 72, 175 U.S.P.Q. at 675 (emphasis added).

21 Yoshida, *supra* note 3, at 463 (calling the Court's use of the term 'algorithm' unfortunate and reasoning that this "imprecision of the Court in characterizing the algorithm before it in *Benson* created uncertainty as to the scope of the exclusionary rule that it upholds and led to many questions in subsequent appeals of rejections by the USPTO about the meaning of the term algorithm") (footnotes omitted).

22 Yoshida, *supra* note 3, at 461 (stating that "[a]ll computer program-related inventions are algorithms").

23 State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1373, 47 U.S.P.Q.2d (BNA) 1596, 1600 (Fed. Cir. 1998), *cert. denied*, ___ S. Ct. ___, 1999 WL 8601, 67 U.S.L.W. 3302, 67 U.S.L.W. 3327, 67 U.S.L.W. 3431, 67 U.S.L.W. 3436 (U.S. Jan 11, 1999).

24 *See Gottschalk*, 409 U.S. at 72, 175 U.S.P.Q. at 676 (modified by author).

25 Parker v. Flook, 437 U.S. 584, 585, 198 U.S.P.Q. (BNA) 193, 195 (1978).

26 *Id.*

27 *Id.*

28 *Id.*

[29] *Id.*

[30] *Id.* at 585-86, 198 U.S.P.Q. at 195.

[31] *Id.* at 586, 198 U.S.P.Q. at 195.

[32] *Id.* at 596-97, 198 U.S.P.Q. at 200.

[33] *See id.* at 588, 198 U.S.P.Q. at 196.

[34] *Id.*

[35] *Id.*

[36] *Id.* at 589-90, 198 U.S.P.Q. at 197.

[37] *Id.* at 590, 198 U.S.P.Q. at 197.

[38] *See id.*

[39] *Id.*

[40] *Id.*

[41] *Id.* at 594, 198 U.S.P.Q. at 199. A previous Supreme Court case treated a mathematical formula as prior art. *See* O'Reilly v. Morse, 56 U.S. 62 (1854) (citing Neilson v. Harford, 151 Eng. Rep. 1266 (1841)).

[42] *Id.* at 596, 198 U.S.P.Q. at 200.

[43] *Id.* at 585, 198 U.S.P.Q. at 195.

[44] Diamond v. Diehr, 450 U.S. 175, 191, 209 U.S.P.Q. (BNA) 1, 10 (1981) (referring to the claims in *Parker*).

[45] *Id.*

[46] *Parker*, 437 U.S. at 586, 198 U.S.P.Q. at 195.

[47] *Id.*

[48] *See id.*

49 *See* STOBBS, *supra* note 2, at 296 (advising to "[m]ake sure the specification contains a full disclosure, not only of the principle or formula involved, but of how to apply the formula or principle to a specific problem").

50 *Id.*

51 450 U.S. 175, 191-93, 209 U.S.P.Q. (BNA) 1, 10 (1981).

52 *Id.* at 177, 209 U.S.P.Q. at 4.

53 *See id.* at 177-78, 209 U.S.P.Q. at 4.

54 *Id.* at 178, 209 U.S.P.Q. at 4.

55 *Id.*

56 *See id.*, 209 U.S.P.Q. at 4-5.

57 *See id.* at 178-79, 209 U.S.P.Q. at 4-5.

58 *See id.* at 179, 209 U.S.P.Q. at 5.

59 *See id.*

60 *Id.*

61 Claims 2 and 11 are as follows:
2. The method of claim 1 including measuring the activation energy constant for the compound being molded in the press with a rheometer and automatically updating said data base within the computer in the event of changes in the compound being molded in said press as measured by said rheometer.
11. A method of manufacturing precision molded articles from selected synthetic rubber compounds in an openable rubber molding press having at least one heated precision mold, comprising:
(a) heating said mold to a temperature range approximating a predetermined rubber curing temperature,
(b) installing prepared unmolded synthetic rubber of a known compound in a molding cavity of predetermined geometry as defined by said mold,
(c) closing said press to mold said rubber to occupy said cavity in conformance with the contour of said mold and to cure said rubber by transfer of heat thereto from said mold,
(d) initiating an interval timer upon the closure of said press for monitoring the elapsed time of said closure,
(e) heating said mold during said closure to maintain the temperature thereof within said range approximating said rubber curing temperature,
(f) constantly determining the temperature of said mold at a location closely adjacent said cavity thereof throughout closure of said press,
(g) repetitively calculating at frequent periodic intervals throughout closure of said press the Arrhenius equation for reaction time of said rubber to determine total required cure time v as follows:
$\ln v = cz+x$
wherein c is an activation energy constant determined for said rubber being molded and cured in said press, z is the temperature of said mold at the time of each calculation of said Arrhenius equation, and x is a constant which is a function of said predetermined

geometry of said mold,
(h) for each repetition of calculation of said Arrhenius equation herein, comparing the resultant calculated total required cure time with the monitored elapsed time measured by said interval timer,
(i) opening said press when a said comparison of calculated total required cure time and monitored elapsed time indicates equivalence, and
(j) removing from said mold the resultant precision molded and cured rubber article.
Id. at 179-80 n.5, 209 U.S.P.Q. at 5 n.5.

62 *Id.* at 179 n.5, 209 U.S.P.Q. at 5 n.5.

63 *See id.* at 179-80, 209 U.S.P.Q. at 5.

64 *See id.* at 181, 209 U.S.P.Q. at 5-6.

65 *See id.* at 181, 209 U.S.P.Q. at 5 (referring to *In re* Diehr, 602 F.2d 982, 203 U.S.P.Q. (BNA) 44 (C.C.P.A. 1979)).

66 *Id.* at 182, 209 U.S.P.Q. at 6 (quoting S. Rep. No. 1979, 82d Cong., 2d Sess., 5 (1952) and Diamond v. Chakrabarty, 447 U.S. 303, 309, 206 U.S.P.Q. (BNA) 193, 197 (1980)). *Diamond v. Chakrabarty* was decided after *Gottschalk* and *Parker*, but before *Diamond v. Diehr*. In *Chakrabarty*, the Supreme Court held that a human-made microorganism was patentable subject matter. *Diamond v. Chakrabarty*, 447 U.S. at 309, 206 U.S.P.Q. at 197. This case signaled the Court's willingness to expand its view of patentable subject matter to "anything under the sun that is made by man." *Id.*

67 *Diehr*, 450 U.S. at 183, 209 U.S.P.Q. at 6 (citing Cochrane v. Deener, 94 U.S. 780, 787-88 (1877)).

68 Cochrane, 94 U.S. at 787-88.

69 *Diehr*, 450 U.S. at 184, 209 U.S.P.Q. at 7 (quoting *Gottschalk*, 409 U.S. at 70, 175 U.S.P.Q. at 676).

70 *Id*

71 *Id.*

72 *Id.* at 185, 209 U.S.P.Q. at 7.

73 *Id.*

74 *Id.*, 209 U.S.P.Q. at 7-8 (citations omitted).

75 *Id.* at 187, 209 U.S.P.Q. at 8.

76 *Id.*

77 *Id.*

[78] *Id.*

[79] *Id.*

[80] *Id.* (emphasis added).

[81] *Id.* (emphasis in original).

[82] *Id.* at 192, 209 U.S.P.Q. at 10.

[83] *Gottschalk v. Benson*, 409 U.S. 63, 67, 175 U.S.P.Q. (BNA) 673, 675 (1972).

[84] *See* discussion *supra* Parts II.A.2. *Parker v. Flook* and II.A.3. *Diamond v. Diehr*.

[85] *See* discussion *supra* Part II.A. Supreme Court Precedent.

[86] *Id.*

[87] *See* discussion *supra* Part II.A.3. *Diamond v. Diehr*.

[88] *Id.*

[89] *See* Burtis, *supra* note 4, at 1137 (refering to the Supreme Court cases of *Gottschalk v. Benson, Parker v. Flook*, and *Diamond v. Diehr* as a flawed trilogy).

[90] Yoshida, *supra* note 3, at 458 (identifying "the explosion of microprocessor and computer technology and the accompanying increase in the importance of the methodology of programming such computers").

[91] 33 F.3d 1526, 31 U.S.P.Q.2d (BNA) 1545 (Fed. Cir. 1994).

[92] *Id.* at 1537, 31 U.S.P.Q.2d at 1551.

[93] *See id.*, 31 U.S.P.Q.2d at 1552.

[94] *See id.*

[95] *Id.* (emphasis in original).

[96] *Id.* at 1538-39, 31 U.S.P.Q.2d at 1553.

[97] *Id.* at 1539, 31 U.S.P.Q.2d at 1553.

98 *See id.*

99 *See id.* at 1539-40, 31 U.S.P.Q.2d at 1553.

100 *Id.* at 1539, 31 U.S.P.Q.2d at 1553 (citations omitted).

101 *Id.*

102 *See id.* at 1539-40, 31 U.S.P.Q.2d at 1554.

103 *Id.* at 1540, 31 U.S.P.Q.2d at 1554.

104 *But see* Lawrence Kass, Comment, *Computer Software Patentability and the Role of Means-plus-function Format in Computer Software Claims*, 15 PACE L. REV. 787, 863 (1995) (alleging that "it is questionable whether the bracketed phrases did actually represent true structure" and stating that "[t] he only bracketed phrase that arguably called for a discreet structural element was … 'a read only memory (ROM)'").

105 *Alappat*, 33 F.3d at 1541, 31 U.S.P.Q.2d at 1555 (emphasis in original).

106 *Id.* at 1541-42, 31 U.S.P.Q.2d at 1555-56.

107 *Id.* at 1542, 31 U.S.P.Q.2d at 1555.

108 *Id.*, 31 U.S.P.Q.2d at 1556 (citing Diamond v. Diehr, 450 U.S. 175, 185, 209 U.S.P.Q. (BNA) 1, 10 (1981)).

109 *See id.* at 1543, 31 U.S.P.Q.2d at 1556.

110 *Id.*, 31 U.S.P.Q.2d at 1556-57 (emphasis in original).

111 *Id.* at 1544, 31 U.S.P.Q.2d at 1557.

112 *Id.*

113 *Id.*, 31 U.S.P.Q.2d at 1558.

114 *Id.*

115 *Id.*

116 *Id.* at 1544-45, 31 U.S.P.Q.2d at 1558 (quoting *Ex Parte* Alappat, 23 U.S.P.Q.2d (BNA) 1340, 1345 (B.P.A.I. 1992)).

[117] *Id.* at 1545, 31 U.S.P.Q.2d at 1558.

[118] *Id.*

[119] *Id.*

[120] *Id.*

[121] *In re* Warmerdam, 33 F.3d 1354, 1355, 31 U.S.P.Q.2d (BNA) 1754, 1755 (Fed. Cir. 1994).

[122] *Id.*

[123] *Id.*

[124] *Id.* at 1355-56, 31 U.S.P.Q.2d at 1755-56.

[125] *See id.*

[126] *Id.*

[127] *Id.* at 1356, 31 U.S.P.Q.2d at 1756.

[128] *Id.* at 1355, 31 U.S.P.Q.2d at 1755.

[129] *Id.* at 1356, 31 U.S.P.Q.2d at 1756.

[130] *Id.*

[131] *Id.* at 1357-58, 31 U.S.P.Q.2d at 1756-57.

[132] *Id.* at 1358, 31 U.S.P.Q.2d at 1757.

[133] *Id.*

[134] *Id.*

[135] *In re* Freeman, 573 F.2d 1237, 197 U.S.P.Q. (BNA) 464 (C.C.P.A. 1978).

[136] *In re* Walter, 618 F.2d 758, 205 U.S.P.Q. (BNA) 397 (C.C.P.A. 1980).

[137] *In re* Abele, 684 F.2d 902, 214 U.S.P.Q. (BNA) 682 (C.C.P.A. 1982).

[138] *Warmerdam*, 33 F.3d at 1359, 31 U.S.P.Q.2d at 1758. However, the M.P.E.P. states that the "*Freeman-Walter-Abele* test may additionally be relied upon in analyzing claims directed solely to a process for solving mathematical algorithm." United States Patent & Trademark Office, 1998 Manual of Patent Examining Procedure, Seventh Edition, § 2106 at 2100-5 (*hereinafter* "M.P.E.P."). Therefore, using the *Freeman-Walter-Abele* test may be appropriate in explaining an invention to an examiner. A 1998 decision by the Federal Circuit, *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368, 1373, 47 U.S.P.Q.2d (BNA) 1596, 1600 (Fed. Cir. 1998), *cert. denied*, ___ S. Ct. ___, 1999 WL 8601, 67 U.S.L.W. 3302, 67 U.S.L.W. 3327, 67 U.S.L.W. 3431, 67 U.S.L.W. 3436 (U.S. Jan 11, 1999), casts additional doubt on the importance of the *Freeman-Walter-Abele* test. The reference to this test may be removed from the M.P.E.P.

[139] *Warmerdam*, 33 F.3d at 1359, 31 U.S.P.Q.2d at 1758.

[140] *Id.* There may have been other reasons why this test was abandoned. *See* Nellie A. Fisher, Comment, *The Patent Eligibility of Computer Implemented Processes in the Wake of In re Alappat: the Diehr Standard Resurrected*, 32 HOUS. L. REV. 517, 552 (1995) (noting that the abandonment of the *Freeman-Walter-Abele* test "and the adoption of the Diehr method of analysis can be justified in light of the limitations imputed into the means-plus-function elements of the claim because the two part test was based upon the breadth given to means-plus-function elements").

[141] *Warmerdam*, 33 F.3d at 1360, 31 U.S.P.Q.2d at 1759.

[142] *Id.*

[143] *Id.*

[144] *Id.* at 1361, 31 U.S.P.Q.2d at 1759.

[145] *Id.* at 1359, 31 U.S.P.Q.2d at 1758.

[146] *Id.* at 1360, 31 U.S.P.Q.2d at 1759 (emphasis added).

[147] *Id.* at 1361-62, 31 U.S.P.Q.2d at 1759-60.

[148] *Id.* at 1362, 31 U.S.P.Q.2d at 1760.

[149] *See id.* at 1358, 31 U.S.P.Q.2d at 1757.

[150] *Id.*

[151] *In re* Lowry, 32 F.3d 1579, 1580, 32 U.S.P.Q.2d (BNA) 1031, 1032 (Fed. Cir. 1994).

[152] *Id.* at 1581, 32 U.S.P.Q.2d at 1033.

153 *Id.*

154 *See id.* at 1581-82, 32 U.S.P.Q.2d at 1033.

155 *Id.*

156 *See id.* at 1582, 32 U.S.P.Q.2d at 1033.

157 *Id.*

158 *Id.*

159 *Id.*

160 *Id.*

161 *Id.*, 32 U.S.P.Q.2d at 1033-34.

162 *Id.*, 32 U.S.P.Q.2d at 1033 (citing *In re* Gulack, 703 F.2d 1381, 1385 n.8, 217 U.S.P.Q. (BNA) 401, 403 n.8 (Fed. Cir. 1983)).

163 *Id.*

164 *Id.* at 1583, 32 U.S.P.Q.2d at 1034 (quoting *In re* Bernhart, 417 F.2d 1395, 1399, 163 U.S.P.Q. (BNA) 611, 615 (C.C.P.A. 1969)).

165 *Id.*

166 *Id.*

167 *Id.*

168 *Id.*

169 *Id.*

170 *Id.* at 1584, 32 U.S.P.Q.2d at 1035.

171 *Id.*

[172] *Id.*

[173] State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 1373, 47 U.S.P.Q.2d (BNA) 1596, 1600 (Fed. Cir. 1998), *cert. denied*, ___ S. Ct. ___, 1999 WL 8601, 67 U.S.L.W. 3302, 67 U.S.L.W. 3327, 67 U.S.L.W. 3431, 67 U.S.L.W. 3436 (U.S. Jan 11, 1999).

[174] *See* U.S. Patent No. 5,193,056, Abstract.

[175] *State St.*, 149 F.3d at 1370, 47 U.S.P.Q.2d at 1598.

[176] *Id.* at 1371-72, 47 U.S.P.Q.2d at 1599.

[177] *Id.* at 1370, 47 U.S.P.Q.2d at 1598.

[178] *Id.*

[179] *Id.*

[180] *Id.*

[181] *Id.*

[182] *Id.* at 1377, 47 U.S.P.Q.2d at 1604.

[183] *Id.* at 1373, 47 U.S.P.Q.2d at 1601.

[184] *Id.*

[185] *Id.* at 1375, 47 U.S.P.Q.2d at 1602.

[186] For a discussion of the *Freeman-Walter-Abele* test, see *supra* Part II.B.2. *In re Warmerdam*.

[187] *State St.*, 149 F.3d at 1373, 47 U.S.P.Q.2d at 1601.

[188] *Id.* at 1374, 47 U.S.P.Q.2d at 1601.

[189] *Id.* at 1375, 47 U.S.P.Q.2d at 1601.

[190] *Id.*, 47 U.S.P.Q.2d at 1603.

[191] *See* discussion *supra* Part II.B.1. *In re Alappat; see also* 35 U.S.C.A. § 112 para. 6 (1994 & Supp. 1998).

[192] *See* discussion *supra* Part II.B.1. *In re Alappat*.

[193] *Id.*

[194] *See* discussion *supra* Parts II.B.1. *In re Alappat* and II.B.2. *In re Warmerdam*.

[195] *See* discussion *supra* Part II.B.4. *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*

[196] *See* discussion *supra* Parts II.B.1. *In re Alappat* and II.B.2. *In re Warmerdam*.

[197] *Id.*

[198] *See* discussion *supra* Part II.B.1. *In re Alappat*.

[199] *Id.*

[200] *See* discussion *supra* Part II.B.3. *In re Lowry*.

[201] M.P.E.P. § 2106 at 2100-4.

[202] *Id.* at § 2106 at 2100-4.

[203] *Id.* at 2100-4 to 2100-5.

[204] *Id.* at 2100-4.

[205] STOBBS, *supra* note 2, § 8.21 at 65-66, 67 (Supp. 1997).

[206] 42 F.3d 1376, 33 U.S.P.Q.2d (BNA) 1194 (Fed. Cir. 1994).

[207] *Id.* at 1377, 33 U.S.P.Q.2d at 1194.

[208] *Id.* at 1378, 33 U.S.P.Q.2d at 1195.

[209] *Id.* at 1378-79, 33 U.S.P.Q.2d at 1196.

[210] *Id.* at 1383, 33 U.S.P.Q.2d at 1200.

[211] *In re* Trovato, 60 F.3d 807, 807, 35 U.S.P.Q.2d (BNA) 1570, 1571 (Fed. Cir. 1995), *vacating In re* Trovato, 42 F.3d 1376, 33 U.S.P.Q.2d (BNA) 1194 (Fed. Cir. 1994). One commentator has opined that the Court vacated its earlier decision because it cast doubt on *Alappat* and the pressure to keep *Alappat* as strong precedent was great. Jonathan E. Retsky, *Computer Software Protection in 1996: a Practitioner's Nightmare*, 29 J. MARSHALL L. REV. 853, 860 (1996) (opining that "[o]ne can only assume that the panel of dissenters [[[[(the panel that decided *Trovato* consisted of the dissenting judges in *Alappat*)] ultimately bowed to internal pressure from their majority brethren to abide by the ruling of the *Alappat* decision and not to cast doubt upon it").

[212] *In re* Trovato, 60 F.3d at 808, 35 U.S.P.Q.2d at 1571 (vacating the earlier *In re* Trovato decision).

[213] M.P.E.P. § 2106 at 2100-5.

[214] *Id.*

[215] *Id.*

[216] *Id.* at 2100-6.

[217] *Id.*

[218] *Id.* (emphasis in original).

[219] *Id.* (emphasis in original).

[220] *Id.* at 2100-7.

[221] *Id.*

[222] *Id.*

[223] *Id.*

[224] *Id.*

[225] *Id.* (emphasis in original).

[226] *Id.*

[227] *Id.*

[228] *Id.*

[229] *Id.*

[230] *Id.* at 2100-7 to 2100-8.

[231] *Id.* at 2100-9.

[232] *Id.* (emphasis in original).

[233] *Id.*

[234] *Id.*

[235] *Id.* at 2100-10 (emphasis in original).

[236] *Id.*

[237] *Id.*

[238] *Id.*

[239] *Id.*

[240] "When nonfunctional 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 nonfunctional descriptive material stored in a computer-readable medium does not make it statutory." *Id.* at 2100-10.

[241] *Id.*

[242] *Id.*

[243] *Id.*

[244] *Id.* at 2100-11.

[245] *Id.*

[246] A computer listing is simply a printout of the source code for a computer program.

[247] M.P.E.P. § 2106 at 2100-11.

[248] *Id.*

[249] *Id.*

[250] *Id.*

[251] *Id.*

[252] *Id.*

[253] *Id.* at 2100-12.

[254] *Id.*

[255] *Id.*

[256] *Id.*

[257] *See id.* at 2100-12 (stating that "[i]f a product claim encompasses any and every computer implementation of a process, when read in light of the specification, it should be examined on the basis of the underlying process") (first emphasis in original, second emphasis added).

[258] *Id.* at 2100-12 to 2100-13 (emphasis in original).

[259] *Id.*

[260] *Id.*

[261] *Id.*

[262] *Id.* at 2100-14.

[263] *Id.*

[264] *Id.*

[265] *Id.*

[266] *Id.*

[267] *Id.* at 2100-15.

[268] *Id.*

[269] *Id.*

[270] *Id.*

[271] *Id.*

[272] *Id.*

[273] *Id.* (citation omitted).

[274] *Id.* (citations omitted).

[275] *Id.* at 2100-15 to 2100-16 (stating, after each example, the following "[t]his process has real world value in …," "[t]he real world value of the invention lies in …," and "[g]eophysical exploration of formations below the surface of the earth has real world value").

[276] *Id.* at 2100-16.

[277] *Id.*

[278] *Id.*

[279] *Id.*

[280] *Id* at 2100-17.

[281] *Id.*

[282] *Id.*

[283] *Id.* at 2100-18.

[284] *Id.*

[285] *Id.*

[286] *Id.*

[287] *Id.*

[288] *Id.*

[289] *Id.* at 20-19 (citing Parker v. Flook, 437 U.S. 584, 585, 198 U.S.P.Q. (BNA) 193, 195 (1978)).

[290] *Id.* (citing *In re* Walter, 618 F.2d 758, 770, 205 U.S.P.Q. (BNA) 397, 409 (C.C.P.A. 1980)).

[291] *Id.* (citing *In re* De Castelet, 562 F.2d 1236, 1244, 195 U.S.P.Q. (BNA) 439, 446 (C.C.P.A. 1977)).

[292] 35 U.S.C. § 112, para. 2 (1994 & Supp. 1998).

[293] M.P.E.P. § 2106 at 2100-20.

[294] *Id.*

[295] *Id.*

[296] *Id.*

[297] *Id.*

[298] 35 U.S.C. § 112, para. 1 (1994 & Supp. 1998).

[299] M.P.E.P. § 2106 at 2100-21.

[300] *Id.* (emphasis in original).

[301] *Id.* at 2100-22.

[302] *Id.*

[303] *Id.* at 2100-5.

[304] *Id.*

[305] *Id.* at 2100-10.

[306] *Id.*

[307] *Id.*

308 *Id.*

309 *Id.* at 2100-15.

310 *Id.*

311 *Id.*

312 Melvin C. Garner *et al., Advanced Claim Drafting and Amendment Writing Workshop for Electronics and Computer-Related Subject Matter, in* ADVANCED CLAIM AND AMENDMENT WRITING 1996, at 227, 281 (PLI Sixth Annual Patent Prosecution Workshop Course Book, 1996) (hereinafter "Garner").

313 STEPHEN A. BECKER, PATENT APPLICATIONS HANDBOOK § 2.03[2] at 2-35 (1995) (hereinafter "BECKER").

314 Garner, *supra* note 312, at 282 (stating that "[s]ome practitioners prefer to draft the specification prior to drafting the claims" because they feel "that drafting the specification first gives them a better understanding of the invention so that they are then in a better position to draft claims").

315 *See id.*

316 *Id.*

317 *Id.* (stating that the "generally preferred approach is to draft at least a few broad claims before embarking on the details of the disclosure"); BECKER, *supra* note 313, § 2.02[5][a] at 2-33.

318 *See* Garner, *supra* note 312, at 282 (stating that "[d]rafting the claims first also helps assure that the terms used in the claims have antecedent basis and support in the specification").

319 BECKER, *supra* note 313, § 2.02[5][a] at 2-33.

320 Garner, *supra* note 312, at 283 (stating that "[t]he unfolding of the detailed disclosure routinely suggests additional subject matter for the claims").

321 I say dependent because at this stage only small differences will probably be added. If a big change occurs in the claims, the claims may not have been "right" in the first place.

322 *See* STOBBS, *supra* note 2, § 7.16 at 238-39 (giving explanation of claim diagrams and also illustrating an example of a computer-implemented invention's claim diagram).

323 Garner, *supra* note 312, at 279 (stating that it is prudent "to present a spectrum of claims from the broadest claim possible considering the prior art to the narrowest claim which appears commercially significant, i.e., a 'picture claim'"); BECKER, *supra* note 313 § 2.02[3][c][ii] at 2-27 (stating that "[t]o be on the safe side, it is important to include at least one relatively narrow claim set, composed only of structure limitations, that will cover the commercial embodiment of the invention"); STOBBS, *supra* note 2, at 244 (comparing a patent with only one claim as "putting all of your eggs in one basket," and asserting that "having at least one broad claim is certainly desirable, it is far better to cover the spectrum with broad, intermediate, and narrow claims").

[324] Garner, *supra* note 312, at 301 (stating that "[p]ower supplies and similar elements used with the invention, but not forming an essential part, should generally not be made an element of a broad claim").

[325] BECKER, *supra* note 313, § 2.03[1] at 2-34

[326] *See id.*, § 2.02[3][c][ii] at 2-27 (stating that "it is not advisable to have only means-plus-function claims in an application; it is better practice to incorporate a variety of different claim formats").

[327] Id., § 2.03[1] at 2-34.

[328] Garner, *supra* note 312, at 303 (maintaining that relying on a single independent claim "can result in failure to claim nuances of the overall combination … and in a failure to claim important subcombinations").

[329] *Id.*

[330] Yoshida, *supra* note 3, at 468.

[331] *See* BECKER, *supra* note 313, § 2.03[1] at 2-34 (stating that "[m] ethod claims need not be tied as closely as apparatus claims to disclosed structure, since method claim elements are intended to define operational steps, not hardware").

[332] *See* Garner, *supra* note 312, at 284 (commenting that "in dealing with inventions where the function is more significant than a specific 'structural' implementation, a functional approach works best").

[333] *See*, BECKER, *supra* note 313, § 2.03[1] at 2-34 (advising that if "the disclosed hardware is nothing more than a conventional computer, then novelty lies in the program alone. Hence, the method claims will be dominant").

[334] *See* Yoshida, *supra* note 3, at 495 (stating that "[c]laims that embrace software standing alone as a process can be proper subject matter, but it is wise to tie the process to some means for carrying out the functions performed by the program").

[335] Garner, *supra* note 312, at 261-62 (enunciating common principles emerging from the 1994 Federal Circuit decisions and the PTO Guidelines as including the principle that "apparatus claims remain more likely to pass muster, as indicated by the summary approval of the apparatus claim in Warmerdam, and the willingness of some panels to find disclosed structure more readily applicable (and limiting) with respect to apparatus than method claims").

[336] *See* David L. Bohan, Note, *Computer Programs: Abstract Ideas or Patentable Subject Matter?*, 29 SUFFOLK U. L. REV. 809, 833 (1995) (stating that "a wise patent prosecutor should draft computer program claims as apparatus claims in means-plus-function language").

[337] *In re* Alappat, 33 F.3d 1526, 1540-41, 31 U.S.P.Q.2d (BNA) 1545, 1554-55 (Fed. Cir. 1994).

[338] Garner, *supra* note 312, at 284.

[339] *Id.* at 284-85 (maintaining that a benefit from simply adding "means for" before the steps in a method claim to obtain an apparatus claim is that this method creates mirror images between the method claims and apparatus claims which "preclude[s] a requirement for restriction between the apparatus and method" (citing M.P.E.P. § 809.03)).

340 *Id.* at 286 (stating that "[a]lthough there is nothing wrong with using means clauses for any of the elements of a computer program-related invention … non-means-plus-function language (i.e., structural language) for at least one of the elements helps define patentable subject matter").

341 Yoshida, *supra* note 3, at 500 ("two sets of system and/or apparatus claims [should] be drafted. The first set should use the means-plus-function form under 35 U.S.C. § 112 para. 6. The second set should use specific architecture and/or hardware components." (citing Robert Greene Sterne, et al., *Preparing and Prosecuting Electronic and Computer Related Applications: Avoiding and Overcoming Statutory Subject Matter Rejections*, 33 J.L. & TECH. 297, 311 (1993))).

342 Garner, *supra* note 312, at 286.

343 35 U.S.C. § 112 para 6 (1994 & Supp. 1998).

344 Garner, *supra* note 312, at 286-87 (stating that where an infringing element is an equivalent to the means-plus-function element's structure in the specification, "literal infringement will be found").

345 *Id.* at 295 (stating that "it is probably preferable to use terms such as 'amplifier' and 'oscillator', for example, rather than the terms 'means for oscillating' … and 'means for amplifying"' because 'amplifier' "would theoretically cover any structure which includes an amplifier" whereas 'means for amplifying', for literal infringement, "would require finding that the alleged infringing amplifier was the structural equivalent of the specific amplifier disclosed in the specification").

346 *See* Kass, *supra* note 104, at 866 (asserting "that means-plus-function format limits the scope of the claims to only those structures that are described in the specification, and a potentially narrow range of equivalents").

347 *See Ex Parte* Stanley, 121 U.S.P.Q. 621 (BNA) (Bd. App. 1958) (finding that "means" was unnecessary where claim recited a jet driving device so constructed and located on the rotor as to drive the rotor); M.P.E.P. § 2181 at 2100-178 (stating that "[a]lthough … there is no particular language that must appear in a claim in order for it to fall within the scope of 35 U.S.C. § 112, sixth paragraph, … it must be clear that the element in the claim is set forth, at least in part, by the function it performs as opposed to the specific structure, material, or acts that perform the function").

348 Garner, *supra* note 312, at 300 (recommending that both means-plus-function and non-means-plus-function limitations be included in a patent application.); BECKER, *supra* note 313, § 2.03[1] at 2-34 (recommending at least one independent claim in means-plus-function format and at least one independent claim in terms of hardware elements).

349 For example, U. S. Patent No. 5,664,191 claim 10 seems to be a method claim disguised as a data structure claim, and U. S. Patent No. 5,652,864 claim 18 seems to be a hybrid between an apparatus claim, a method claim, and a data structure claim.

350 *See* Garner, *supra* note 312, at 262 (stating that "[d]ata structure claims are on somewhat more slippery ice").

351 *Id.*

352 Diana Roberts, *The State of Computer Software Product Claims after In re Alappat*, 2 U. BALT. INTELL. PROP. J. 219, 219-20 (1994).

353 *See* M.P.E.P. § 2106 at 2100-10.

354 *See* Garner, *supra* note 312, at 262 (stating that "[m]ore complete disclosure of physical implications of processing will enhance survival").

[355] 35 U.S.C. § 112 (1994 & Supp. 1998).

[356] For a general treatment of the enablement requirement, *see* 3 DONALD S. CHISUM, CHISUM ON PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY AND INFRINGEMENT § 7.03 (1997).

[357] Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 941, 15 U.S.P.Q.2d (BNA) 1321, 1329 (Fed. Cir. 1990).

[358] *Id.* at 941, 15 U.S.P.Q.2d at 1329 (stating that "[t]he amount of disclosure that will enable practice of an invention that utilizes a computer program may vary according to the nature of the invention, the role of the program in carrying it out, and the complexity of the contemplated programming, all from the viewpoint of the skilled programmer").

[359] *In re* Naquin, 398 F.2d 863, 866, 158 U.S.P.Q. (BNA) 317, 319 (C.C.P.A. 1968).

[360] Garner, *supra* note 312, at 266.

[361] *See id.* at 266-67 (stating the "[i]t is up to the patent draftsman to include sufficient disclosure so that only routine programming efforts remain to be accomplished in order to practice the invention").

[362] *Id.* at 269.

[363] For interesting articles about object-oriented programming techniques and their possible use in patenting software, *see* Keith Stephens & John P. Sumner, *Software Objects: A New Trend in Programming And Software Patents*, 12 SANTA CLARA COMPUTER & HIGH TECH. L.J. 1 (1996) (discussing patent protection for object-oriented software and elaborating on the availability and advantages of patenting software objects as articles of manufacture); Dryja, *supra* note 5, at 143 (discussing how software methodology has had a gradual shift from the Turing model to an object-oriented model where the Turing model was more suited to copyright law protection and the object-oriented model is suited more for patent law protection).

[364] For a nice summary of different ways to describe software, *see generally* GREGORY A. STOBBS, SOFTWARE PATENTS ch. 6, at 183 (1995).

[365] Garner, *supra* note 312, at 270 (stating that at a minimum, "a well drafted application with a computer program involved will contain a plain and concise description of the program in words, including its operation and the environment in which it is used").

[366] *Id.*

[367] *Id.* at 270-71 (remarking that it "is preferable to describe the program in general terms without relating it to a particular programming language" because a more limited description may lead to more limited claims under 35 U.S.C. § 112).

[368] "C" is a common programming language.

[369] *See* Garner, *supra* note 312, at 272 (maintaining that "[f]low charts are probably the program disclosure vehicle of choice, and the failure to have one could easily prompt a § 112 rejection in a computer case"); BECKER, supra note 313, § 1.03[8][d] at 1-38 (stating that patent "applications in computer program-implemented inventions should make liberal use of flow charts to describe the sequence of operations that the program will carry out").

[370] Garner, *supra* note 312, at 273.

[371] *See id.* at 274 (asserting that the "written description should refer to each of the program boxes and the flow of the program from initiation to completion."); BECKER supra note 313, § 1.03[10][b] at 1-48 (stating that "[e] ach element of the flow chart should represent a programming step, routine or subroutine that could have been written by an ordinary programmer without requiring undue experimentation").

[372] *See* Garner, *supra* note 312, at 275 (stating that source code listings "are primarily a relic of the early days of computer program patents when it was unclear what would suffice for sufficiency of disclosure"); BECKER, *supra* note 313, § 1.03[8][d] at 1-38 (stating that "[i]n most cases, it is not necessary or desirable to include a complete program listing in the application"); *Cf.* Fonar Corp. v. General Elec. Co., 107 F.3d 1543, 1548-50, 41 U.S.P.Q.2d (BNA) 1801, 1804-05 (Fed. Cir. 1997) (finding that to satisfy best mode requirement patent needs to describe the functions of the software and that the source code is not required to adequately disclose the functions of the software); Robotic Vision Sys., Inc. v. View Eng'g, 112 F.3d 1163, 1166, 42 U.S.P.Q.2d (BNA) 1619, 1622-23 (Fed. Cir. 1997) (ruling that when the disclosure of software is required to satisfy the best mode requirement, it is generally sufficient if the functions of the software are disclosed because usually creating source code from the software is within the ability of one skilled in the art (citing Fonar Corp. v. General Elec. Co., 107 F.3d at 1549-50, 41 U.S.P.Q.2d at 1805; *In re* Hayes Microcomputer Prods., Inc. Patent Litigation, 982 F.2d 1527, 1537-38, 25 U.S.P.Q.2d (BNA) 1241, 1248-49 (Fed. Cir. 1992)).

[373] For example, readers with some computer science background have undoubtedly read through the source code for a quicksort routine many times, yet likely still can't explain how it works.

[374] BECKER, *supra* note 313, § 1.03[10] at 1-45.

[375] *See* Garner, *supra* note 312, at 275 (commenting that it "is advisable to have some portion of a program listing where there is some aspect of the program that requires other than relatively straightforward programming from the flow chart"); BECKER, *supra* note 313, § 1.03[10][b] at 1-48 (asserting that the source code is a "form of disclosure that can be used safely to describe an unusual program step").

[376] *But see* Retsky, *supra* note 4, at 864 (stating that to be on the safe side, "an applicant should file both the program source code and flowcharts"). "A software patent without source code is like a law review piece filled with case names but missing citations to case reporters. A person of ordinary skill in legal research might be able to track down the full-text of all the opinions. *Marbury v. Madison* would be found quicker than a state trial court opinion. But, would anyone think that such a practice was enabling or the best mode? As it is now, the disclosure requirements can be met using such devices as specifications, flowcharts, and pseudo-code." Burke, *supra* note 4, at 1158 (footnotes omitted).

[377] Garner, *supra* note 312, at 276 (maintaining that "[p]rogram listings are used in only the most unusual of circumstances, e.g., where it would not be clear to a person skilled in the art how a portion of the flow chart would be implemented").

[378] *Id.* at 275 (asserting that "non-program aspects can be illustrated by block diagrams where the blocks represent well known hardware."); BECKER, supra note 313, § 1.03[10][a][i] at 1-46 (stating that "each block of a block diagram must be composed of off-the-shelf circuit components, or the content of the block must be described in detail").

[379] *See* Garner, *supra* note 312, at 275 (stating that hardware "should be identified in the specification by listing the manufacturer and model number, where available"); BECKER, *supra* note 313, § 1.03[10][b] at 1-47 (maintaining that "[i]f the content of a block is a commercial device, the manufacturer and part identification should be given").

[380] BECKER, *supra* note 313, § 1.03[10][b] at 1-47 (asserting that if "the content of the block is not an off-the-shelf device, the structure and function of the component should be shown in a separate figure and described in detail in the specification").

[381] 35 U.S.C. § 112 (1994 & Supp. 1998). For a general treatment of the best mode requirement, *see* 3 DONALD S. CHISUM,

CHISUM ON PATENTS: A TREATISE ON THE LAW OF PATENTABILITY, VALIDITY AND INFRINGEMENT § 7.05 (1997).

382 U.S. Gypsum Co. v. National Gypsum Co., 74 F.3d 1209, 1212, 37 U.S.P.Q.2d (BNA) 1388, 1390 (Fed. Cir. 1996).

383 With many of the configuration management systems being used today, it would be simple for a company to build a specific version of software, or to build the latest software as of a specific date.

384 *See* BECKER, *supra* note 313, § 1.03[13][a] at 1-51 to 1-52 (stating that, for a best mode involving a software implementation, an "examiner should accept disclosure in the form of a flow chart without any disclosure of a specific computer listing as evidence that the best mode has been disclosed").

385 "[A] good case can be made for filing a patent application early in the development project, perhaps when only a 'bare bones' implementation is at hand. In these circumstances, the best mode to be disclosed can be very much less than it might when the project is nearing completion and most of the details of the implementation are complete (often after trying a number of alternatives)." Garner, *supra* note 312, at 277.

386 BECKER, *supra* note 313, § 1.03[13][a] at 1-52.

387 *Id.* (stating that "[i]t is not necessary to disclose the best mode for practicing subject matter not claimed in the application, as unclaimed subject matter does not constitute the invention").

388 *Id.*

389 107 F.3d 1543, 41 U.S.P.Q.2d (BNA) 1801 (Fed. Cir. 1997).

390 Claim 1 for this patent is as follows:
1. A method for obtaining in the course of a single scan NMR image data for a plurality of differently oriented selected planes in an object using nuclear magnetic resonance techniques, said method comprising the steps of:
(a) positioning an object in a static homogeneous magnetic field;
(b) determining first and second selected planes in said object for which NMR image data is to be obtained, said first selected plane being located at a first portion of said object and having a first orientation with respect to a predetermined direction and said second selected plane being located at a second portion of said object and having a second orientation with respect to said predetermined direction, said first and second orientations being different from one another;
(c) subjecting said object to a plurality of repetitions of a first repetition sequence composed of NMR excitation and magnetic gradient field pulses, each of said repetitions of said first repetition sequence including the steps of applying an excitation pulse and reading out of an NMR signal produced by said excitation pulse, said excitation pulse for said first repetition sequence being applied at a first predetermined frequency in the presence of a first predetermined slice selector magnetic field gradient having a gradient direction extending perpendicular to said first selected plane, said first predetermined frequency being chosen so that said application of said excitation pulse at said first predetermined frequency only excites selected nuclei in said first selected plane, and said plurality of repetitions of said first repetition sequence being carried out in a manner to encode spatial information into a first collection of said NMR signals, said first collection of NMR signals being representative of NMR image data for said first selected plane; and
(d) subjecting said object to a plurality of repetitions of a second repetition sequence composed of NMR excitation and magnetic field gradient pulses, each of said repetitions of said second repetition sequence including the steps of applying an excitation pulse and reading out of an NMR signal produced, by said excitation pulse, said excitation pulse for said second repetition sequence being applied at a second predetermined frequency in the presence of a second predetermined slice selector magnetic field gradient having a gradient direction extending perpendicular to said second selected plane, said second predetermined frequency being chosen so that said application of said excitation pulse at said second predetermined frequency only excites selected nuclei in said second selected plane, said second predetermined slice selector magnetic field gradient and said second predetermined frequency being different from said first predetermined slice selector magnetic field gradient and said first predetermined frequency, respectively, and said plurality of repetitions of said second repetition sequence being carried out in a manner to encode spatial information into a second collection of NMR signals, said second collection of NMR signals being representative of NMR image

data for said second selected plane;
said plurality of repetitions of said first and second repetition sequences each being carried out during the course of a single scan of said object and each being continued substantially throughout said single scan, the repetition time interval for repeating each of said first and second repetition sequences being substantially the same and said steps of applying an excitation pulse and reading out of an NMR signal for each repetition of said second repetition sequence being performed at a different time during said repetition time interval than each of said steps of applying an excitation pulse and reading out of an NMR signal for said first repetition sequence.

391 Fonar Corp. v. General Elec. Co., 107 F.3d 1543, 1546, 41 U.S.P.Q.2d (BNA) 1801, 1803 (Fed. Cir. 1997).

392 The LGRAD and GETMAO programs. *Id.* at 1548, 41 U.S.P.Q.2d at 1804.

393 *Id.*

394 *Id.* at 1547, 41 U.S.P.Q.2d at 1804.

395 *Id.*

396 *Id.*

397 *Id.*

398 *Id.* at 1548, 41 U.S.P.Q.2d at 1804.

399 *Id.* at 1549, 41 U.S.P.Q.2d at 1805.

400 *Id.*

401 *Id.*

402 *Id.*

403 *See* U.S. Patent Number 4,871,966.

404 "Function" as used in a computer programming context.

405 *Fonar*, 107 F.3d at 1549, 41 U.S.P.Q.2d at 1805.

406 *Id.* at 1550, 41 U.S.P.Q.2d at 1806.

407 *Id.* at 1549, 41 U.S.P.Q.2d at 1805.

408 *Id.* at 1550, 41 U.S.P.Q.2d at 1806.

409 *Id.*

410 Robotic Vision Sys., Inc. v. View Eng'g, 112 F.3d 1163, 42 U.S.P.Q.2d (BNA) 1619 (Fed. Cir. 1997).

411 *Id.* at 1164, 42 U.S.P.Q.2d at 1620.

412 *Id.*, 42 U.S.P.Q.2d at 1621.

413 Claim 1 read as follows: 1. A method for obtaining three-dimensional data from devices having corresponding sides, comprising the steps of: providing a multi-pocketed tray with tray pockets arranged in rows and columns; scanning sequentially with at least one three-dimensional sensor corresponding sides of said devices in a row or column; and repeating said scanning step for all rows and columns containing sides of said devices from which data is to be obtained.

414 *Robotic Vision Sys.*, 112 F.3d at 1164, 42 U.S.P.Q.2d at 1621.

415 *Id.* View also argued that the patent was invalid because of an alleged violation of the on-sale bar. *Id.*

416 *Id.*

417 *Id.* at 1166, 42 U.S.P.Q.2d at 1622.

418 *Id.*

419 *Id.* (emphasis added).

420 *Id.*

421 *Id.* at 1166-67, 42 U.S.P.Q.2d at 1623.

422 *See* Retsky, *supra* note 4, at 864 (stating that to be on the safe side, "an applicant should file both the program source code and flowcharts").

423 For a criticism of flow diagrams, *see id.* at 864-65 (asserting the flowcharting deserves a just burial "as an obsolete relic in the history of software and computer evolution").

424 A claim was categorized as a data structure claim if the preamble stated "the data structure comprising," or the like, before the elements of the claim were enumerated. Unless it was clearly claiming a data structure, the claim was categorized as either an apparatus claim or a method claim.

425 The length of the claim was determined by obtaining an electronic copy of the claim from Lexis, deleting the claim number, and, using Corel WordPerfect 7's "word count" feature.

426 This number has been rounded to the nearest whole number.

427 The summary may also include the objects of the invention.

428 U. S. Patent No. 5,666,552 claim 6 (emphasis and notations added).

429 U. S. Patent No. 5,670,986 claim 1 (emphasis and notations added).

430 U. S. Patent No. 5,671,446, claim 39 (emphasis and notations added).

431 M.P.E.P. § 2106 at 2100-11.

432 A claim was categorized as a data structure claim if the preamble stated "the data structure comprising," or the like, before the elements of the claim were enumerated. Unless it was clearly claiming a data structure, the claim was categorized as either an apparatus claim or a method claim.

433 The length of the claim was determined by obtaining an electronic copy of the claim from Lexis, deleting the claim number, and, using Corel WordPerfect 7's "word count" feature.

434 This number has been rounded to the nearest whole number.

435 The summary may also include the objects of the invention.

436 U. S. Patent No. 5,594,910, claim 7 (emphasis and notations added).

437 U. S. Patent No. 5,586,296, claim 6 (emphasis and notations added).

438 U. S. Patent No. 5,652,864, claim 18 (emphasis and notations added).

439 A claim was categorized as a data structure claim if the preamble stated "the data structure comprising," or the like, before the elements of the claim were enumerated. Unless it was clearly claiming a data structure, the claim was categorized as either an apparatus claim or a method claim.

440 The length of the claim was determined by obtaining an electronic copy of the claim from Lexis, deleting the claim number, and, using Corel WordPerfect 7's "word count" feature.

441 This number has been rounded to the nearest whole number.

442 The summary may also include the objects of the invention.

443 U.S. Patent No. 5,659,685, claim 12 (emphasis and notations added).

[444] U. S. Patent No. 5,664,133, claim 1 (emphasis and notations added).

[445] U. S. Patent No. 5,664,191, claim 10 (emphasis and notations added).

[446] A claim was categorized as a data structure claim if the preamble stated "the data structure comprising," or the like, before the elements of the claim were enumerated. Unless it was clearly claiming a data structure, the claim was categorized as either an apparatus claim or a method claim.

[447] The length of the claim was determined by obtaining an electronic copy of the claim from Lexis, deleting the claim number, and, using Corel WordPerfect 7's "word count" feature.

[448] U. S. Patent No. 5,596,574, claim 1 (emphasis and notations added).

[449] U. S. Patent No. 5,666,532, claim 29 (emphasis and notations added).

[450] *See* discussion *supra* Part II.B.1. *In re Alappat*.

[451] *Id.*

[452] *See* discussion *supra* Part II.B.4. *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*

[453] *Id.*

[454] *See* discussion *supra* Part II. Subject Matter Case Law.

[455] *Id.*

[456] *See* discussion *supra* Part II. Subject Matter Case Law; Part III. Examination Guidelines for Computer-Related Inventions; and Part IV. Practical Tips in Writing Claims. *See also* GREGORY A. STOBBS, SOFTWARE PATENTS § 9.21, at 296 (1995) (stating that a lesson of *Parker* is "that, when possible, [an author should] avoid claiming a software process or algorithm such that the result of the process is simply a number").

[457] *See* discussion *supra* Part III. Examination Guidelines for Computer-Related Inventions and Part IV. Practical Tips in Writing Claims.

[458] *See* discussion *supra* Part III. Examination Guidelines for Computer-Related Inventions.

[459] *Id.*

[460] *See* discussion *supra* Part IV.A. Draft the Claims First.

[461] *Id.*

[462] *See* discussion *supra* Part IV. Practical Tips in Writing Claims.

[463] *Id.*

[464] *Id.*

[465] *See* discussion *supra* Part IV. Practical Tips in Writing Claims and particularly Part IV.D.3. *In re Lowry*.

[466] *See* discussion *supra* Part II.B.1. *In re Alappat*.

[467] *See* discussion *supra* Part IV. Practical Tips in Writing Claims.

[468] *See* discussion *supra* Part V. The Specification.

[469] *Id.*

[470] *Id.*

[471] *Id.*

[472] *See* discussion *supra* Part V.A. Enablement.

[473] *Id.*

[474] *Id.*

[475] *See* discussion *supra* Part V.B. Best Mode.

[476] *Id.*

[477] *Id.*