

From: Erwin Basinski

Sent: Sunday, February 10, 2013 1:35 AM

To: SoftwareRoundtable2013

Subject: Comments in response to Notice of Roundtable Events regarding Software-Related Patents

Sirs,

I am pleased to submit the attached comments in response to this notice - FR Doc No: 2012-31594.

Respectfully

Erwin J. Basinski

Comments of Erwin J. Basinski in response to
Request for Comments and Notice of Roundtable Events for
Partnership for Enhancement of Quality of Software-Related Patents

United States Patent and Trademark Office
[Docket No. PTO-P-2012-0052] [FR Doc No: 2012-31594]

Background

The following comments on the subject of Software and Computer Implemented Inventions are respectfully submitted. By way of background, I am a Solo Practice Patent Attorney, whose practice has focused on Computer Implemented Inventions for over twenty years. During this time I was an in-house Patent Attorney at Sun Microsystems, an Of Counsel at Morrison Foerster LLP, and a Solo Practice attorney. Prior to starting this practice I was a computer programmer, and manager of programmers for over 20 years, starting with the IBM 650 machine in 1958 at Shell Chemical Company in Texas. I subsequently managed software related projects and programmers at Texaco (early programming in TEXTRAN And FORTRAN), ITT (working on the US Naval Tactical Data System), GE Apollo Support (managing programmers on the Apollo Project), Harris Corporation (Data Center Manager) and Carterfone Corp.

Summary

Based on my background experience, it is my belief that the objectives of the subject Roundtable can be met if a partnership with the “computer related technical” community, and the legal community, could result in an agreement to stop using the misunderstood general term “Software,” and instead use specific item names/words that more particularly describe the item/device at issue: i. e. the computer program, subprogram, tool, compiler, debugger, run-time compiler, operating system program, application program, etc. I firmly believe that doing so would provide the means to enhance the quality of Computer Implemented Patents that conform to the USPTO’s definition of a “quality patent,” which includes providing that (1) the protection granted is of proper scope; and (2) which provides sufficiently clear notice to the public as to what is protected by the claims. The specific comments below provide a description of the confusion resulting from the general usage of the misunderstood term “software;” a recommended framework for using more specific terms to better identify the items at issue, as well as their scope; and some general suggestions for claim drafting which could at least be a basis for a discussion about what types of more specific terms should be considered for the language of claim limitation and description.

Legal Background of Computer Related Patents

It is not appropriate in these comments to review the historical legal background of the patenting of software related patents, or as I prefer to call them, Computer Related

Invention (CII) patents. Those interested in this legal background are referred to the writings and books of Professors Donald Chisum (see www.chisum.com), Mark Lemley, Stanford University Law School, and Robert Merges, Bolt Hall, University of California at Berkeley, among others.

It is necessary however, to review the current state of the law of CII patents to understand the issues of the subject Roundtable discussions on CII patents. The current CII patent landscape revolves around the 2010 Supreme Court decision in the case *Bilski v. Kappos*, (130 S. Ct. 3218 (2010)). Professor Chisum characterized the *Bilski* decision as “a remarkably inconclusive contribution to the law on patent eligible subject matter. The “holdings” in *Bilski* are summarized by Professor Chisum (see <http://www.chisum.com/current-developments/bilski-watch/notes-on-bilski>” as follows:

“In the portions of the *Kennedy* opinion concurred in by the Court majority, the Court held that:

(a) the “three specific exceptions to §101’s broad patent-eligibility principles” derived from case law, to wit: “ ‘laws of nature, physical phenomena, and abstract ideas’,” remain the law even though those exceptions were “not required by the statutory text” (Section 101).

(b) the MORT (machine-or-transformation) test is a “useful and important clue, an investigative tool, for determining whether some claimed inventions are processes under §101,” but, contrary to the Federal Circuit majority, the MORT test “is not the sole test for deciding whether an invention is a patent-eligible ‘process’.”

(c) Section 101’s category of “process,” which Section 100(b) defines as including a “method,” does not “categorically exclude[] business methods,” because (a) the ordinary meaning of “method” does not exclude business methods, (b) it was not clear “how far a prohibition of business method patents would reach,” (c) it was not clear whether such a prohibition “would exclude technologies for conducting a business more efficiently,” and (d) the Section 273 infringement defense clarified that “a business method is simply one kind of ‘method’ that is, at least in some circumstances, eligible for patenting under §101.”

(d) *Bilski*’s “patent application ... falls outside of §101,” not because of either of the two rejected “broad and atextual (sic) approaches,” that is, the MORT test and the business method categorical exclusion, but because its claims cover an abstract idea—to wit, either the “basic of

concept of hedging” (claim 1), that concept reduced to a mathematical formula (claim 4), or that concept limited to a particular field or to “token post solution components,” abstract ideas being one of the Section 101 exceptions established in the Benson, Flook and Diehr decisions.

Concluding, the Court cautioned that it was neither endorsing past Federal Circuit precedent nor precluding the Federal Circuit from developing “other limiting criteria,” including “less extreme means of restricting business method patents.”

Professor Chisum further comments that

“The Court’s characterization of the claims as “abstract ideas” is palpably unsatisfying. The claims were to a series of specified steps a human can take (e.g, “identifying market participants” and “initiating a series of transactions.” The claimed subject matter may have been very obvious in view of the state of the art or possibly unduly vague, but to characterize it as an ”abstract idea” stretches the meaning of “abstract” and “idea” beyond recognition.”

The Bilski Court offers no explanation of what is "an abstract idea" or why the claims at issue "pre-empt" the "concept" of hedging. *Bilski v. Kappos*, 130 S. Ct. at 3235-36 (2009) (Stevens, J., concurring).

Post-Bilski cases, District Court as well as Federal Circuit cases have struggled with the task of trying to define “Abstract Idea,” and when is an “application” of an abstract idea sufficiently described and claimed as to be eligible patent subject matter.

Two such cases dramatically illustrate this struggle at the Federal Circuit. In July 2012, the Federal Circuit issued two decisions regarding patentable subject matter for computer-implemented business methods, *CLS Bank Int’l v. Alice Corp.*, No. 2011-1301 (Fed. Cir. July 9, 2012), and *Bancorp Services v. Sun Life Assurance Co. of Canada*, No. 2011-1467 (Fed. Cir. July 26, 2012). Robert Sachs, Patent Partner at Fenwick & West LLP, has written an article that wonderfully illustrates the conflicts that exist at the Federal Circuit over these issues. See <http://www.fenwick.com/publications/Pages/The-Battle-Rages-On.aspx>. Mr. Sachs describes the conflict as follows:

These two cases “vividly demonstrate the conflicting approaches various members of the court apply to this question. On what appears to be similar business methods patents, the two panels came to opposite decisions — the

CLS panel finding the claims patent-eligible, and the Bancorp panel holding not. The decisions can be reconciled when considered in light of the adequacy of the respective patent disclosures. From a strategic perspective, the decisions reinforce the need for substantively robust specifications that do more than nominally describe the use of a computer in the performance of a business method.”

Mr.Sachs continues

“As expected, both Federal Circuit panels cited the same general precedents of Bilski, Prometheus, Benson, Cybersource, Fort Properties and Dealertrack to frame their analysis. Thus, both panels asserted that claiming "mere computer implementation cannot render an otherwise abstract idea patent eligible" and that the form of the claim, whether a system or computer program product, does not change the "the basic character of a process claim." CLS, slip op. at 22 citing Cybersource, and Bancorp, slip op. at 19 citing Dealertrack.

Most interesting, both panels ultimately relied on the "meaningful limitations" rule set forth in SiRF Tech., Inc. v. Int’l Trade Comm’n, 601 F.3d 1319, 1333 (Fed. Cir. 2010), that addition of machine elements in the claim "impose[s] a meaningful limit on the scope of the claim." The CLS court put special emphasis on SiRF’s statement that a meaningful limit exists when the computer elements do more than function solely as the "obvious mechanism for permitting a solution to be achieved more quickly" by the use of the computer "for performing calculations." CLS, slip op. at 25. The Bancorp court further characterized this rule as requiring that the computer must "facilitat[e] the process in a way that a person making calculations or computations could not," and thus using a computer "only for its most basic function, the performance of repetitive calculations" does not impose a meaningful limitation. Bancorp, slip op. at 19-20.”

Mr. Sachs goes on to explain that claims in CLS were held to be patentable while similar claims in Bancorp were not allowed. The reasons for the different outcomes were arguably because the two panels had different ideas about the adequacy of the description of the computer system and programs in the respective applications as well as different approaches to interpreting the claim language. Thus we arrive at the questions posed by the USPTO for this Roundtable discussion – to wit-

What is required in CII patent applications and claim language to provide that (1) the protection granted is of proper scope; and (2) which provides sufficiently clear notice to the public as to what is protected by the claims.

It is worth noting that on February 8, 2013, the Federal Circuit sitting en banc, heard oral arguments on the CLS Bank v. Alice Corp case. An audio version of this oral argument can be found at <http://www.cafc.uscourts.gov/oral-argument-recordings/search/audio.html> . It is highly recommended that participants listen to this audio before attending the Roundtable meetings. The en banc court will focus on the patenting of inventions implemented through software (CII). The two particular questions highlighted in the en banc order are:

a. What test should the court adopt to determine whether a computer-implemented invention is a patent ineligible "abstract idea"; and when, if ever, does the presence of a computer in a claim lend patent eligibility to an otherwise patent-ineligible idea?; and

b. In assessing patent eligibility under 35 U.S.C. § 101 of a computer-implemented invention, should it matter whether the invention is claimed as a method, system, or storage medium; and should such claims at times be considered equivalent for § 101 purposes?

Regarding this en banc hearing, Dennis Crouch in his daily patent blog (see the Feb. 5th entry at <http://patentlaw.typepad.com/patent/>), indicates that at this oral argument

“In addition to the parties, the Federal Circuit has also granted leave for the USPTO to participate at the oral arguments. The USPTO’s brief focused on practical mechanisms for the process of determining § 101 eligibility. The agency wrote “the essential question under § 101 is whether the claim, properly construed, incorporates enough meaningful limitations to ensure that it amounts to more than a claim for the abstract idea itself . . . This Court should identify a non-exhaustive list of factors for district courts and examiners to consider in resolving that essential question on a case-by-case basis.””

I sincerely doubt that the Federal Circuit will come up with such a “non-exhaustive list of factors.” After listening to the oral argument, it is not clear to me how a majority of the Judges will rule in this case. It did appear that most of the judges agree that in such cases, the patentability (i.e. section 101) test should be something like

- (1) is the claim more than an abstract idea, where an abstract idea is a method that can be done in the head (i.e. purely mental; can be done with pencil & paper); and

(2) if the claim has a limitation which articulates a computer, is the computer playing a significant role (i.e. central to the method claimed) in the process claimed?

If the answer is yes to either question then the claim would be considered as passing the 101 test for patent eligibility.

Also in the oral argument, Judge Moore proffered a rule that “the court must look at each claim separately (i.e. not consider the method claim, the apparatus/system claim and the storage medium claim, as the same for purposes of the above test). However it was not clear that most of the Judges agreed with this suggestion.

It is therefore in this landscape that the subject Roundtable explores these issues and that these comments that follow are directed.

Some additional background - What is “Software?”

In the early days of commercial computers (1950s), when one bought or leased a computer system, they received a set of Hardware (computer processor, storage unit, printer, cables, etc.) and a set of Software (operating system (OS) program in some form, perhaps a listing of the OS program, an operating instruction manual, perhaps a maintenance manual, a programming guide, etc.). Generally the word Software was taken to mean “anything that is not hardware.” Subsequently the “set of software” came to include (i.e. one could also buy/lease additional types of programs from the equipment manufacturers or 3rd parties) such computer programs as compilers, debuggers, data base programs, communication programs, and ultimately various application programs such as accounting systems, payroll systems, etc.

This general definition of “Software” has persisted. For example:

The Merriam-Webster Dictionary current defines Software as follows:

“Definition of SOFTWARE

: something used or associated with and usually contrasted with hardware: as

a : the entire set of programs, procedures, and related documentation associated with a system and especially a computer system; specifically : computer programs;

b : materials for use with audiovisual equipment.”

Wikipedia has a much more extensive exploration/definition of the word Software, which I will not quote in its entirety, but which partially says

“Computer software is so called to distinguish it from computer hardware, which encompasses the physical interconnections and devices required to store and execute (or run) the software.”

Wikipedia, moreover adds to the ambiguity of what software is by in some cases, narrowing the definition

“Computer software, or just software, is a collection of computer programs and related data that provides the instructions for telling a computer what to do and how to do it. Software refers to one or more computer programs and data held in the storage of the computer”

while at the same time expanding the definition to include other concepts

“In other words, software is a set of programs, procedures, algorithms and its documentation concerned with the operation of a data processing system.”

This equating of the term “Software” with “procedures” and/or “algorithms” further adds to the garbled nature of the term software, which both the Courts and the Law struggle to understand, not only in the United States but worldwide.

Thus the word “software” is the epitome of a polysemous word (i.e. a word, phrase, or sign that has multiple related meanings (sememes)).

Some legal definitions that have arisen in various US Court Cases include:

Synercom Tech. v. University Computing Co.
462 F. Supp. 1003 - Dist. Court, ND Texas, 1978 –

“... In the industry, the physical machinery is referred to as hardware and the instructional material as software. ... 1006 McAuto engaged in the "software" end of the computer business, as later did Synercom and EDI, a defendant. ...”

U. S. Supreme Court cases defining software include the following:

Parker v. Flook, 437 US 584 - Supreme Court 1978

Footnote “[7] The term "software" is used in the industry to describe computer programs. The value of computer programs in use in the United States in 1976 was placed at \$43.1 billion, and projected at \$70.7 billion by 1980 according to one industry estimate. See Brief for the

Computer & Business Equipment Manufacturers Assn. as Amicus Curiae 17-18, n. 16.”

And

Microsoft Corp. v. AT&T Corp, 550 U.S. 437 (2007),

the term "software" refers generally to the "set of instructions, known as code, that directs a computer to perform specified functions or operations." Id at 447.

Thus we have multiple meanings of the same word even there. Accordingly, it makes no sense to speak of “software” inventions, or the patentability of “software,” or that “software” is an “algorithm” or procedure, because of the garbled and ambiguous nature of the term “software.”

It is worth noting that the European Union as well as European Courts have struggled with this ambiguous definition of the term “software.” In a recent book “Patent Law for Computer Scientists” written by three Patent Examiners at the European Patent Office and published by Springer in 2010, the authors write at page 20

“The term “software patent” is usually avoided by those discussing the applications implicated because “software” does not have a universally accepted definition. . . . Software is frequently understood to be a set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system or to be information that may provide instructions for computers. . . . To avoid the wrong perception, namely that in general computer code can be patented, the term “software” is used with caution in the context of this book.”

The authors focus on the phrase “Computer Implemented Invention (CII)” when talking about patents in this area.

Interestingly, the Supreme Court, in the Bilski case (130 S. Ct. 3218 (2010)) used the term “software” only two times in the opinion:

“As numerous amicus briefs argue, the machine-or-transformation test would create uncertainty as to the patentability of software, advanced diagnostic medicine techniques, and inventions based on linear programming, data compression, and the manipulation of digital signals.” At page 9, and

“Footnote 40 Forty years later, Judge Rich authored the State Street opinion that some have understood to make business methods patentable. But State Street dealt with whether a piece of software could be patented and addressed only claims directed at machines, not processes. His opinion may therefore be better understood merely as holding that an otherwise patentable process is not unpatentable simply because it is directed toward the conduct of doing business—an issue the Court has no occasion to address today. See State Street, 149 F. 3d, at 1375.” At page 31.

It seems clear from the above, that legally, what we are talking about are “computer programs” that are defined as the "set of instructions, known as code, that directs a computer to perform specified functions or operations."

The US Patent & Trademark Office (PTO), in its reply Brief to the US Supreme Court in the Bilski case (<http://www.patentlyo.com/08-964bsunitedstates.pdf>), in discussing the term "software" at page 37-38, says

"In any event, the machine-or-transformation test contemplates that many forms of "software" inventions are patent-eligible. As with many types of technology, the patent-eligibility of software cannot be resolved as a categorical matter. Rather, the eligibility of a claimed software invention depends on the content of that invention and the form in which it is sought to be patented"

The brief then points to footnote 11 which says

"As this court [i.e. the Supreme Court] recently observed in *Microsoft Corp. v. AT&T Corp*, 550 U.S. 437 (2007), the term "software" refers generally to the "set of instructions, known as code, that directs a computer to perform specified functions or operations." Id at 447. A claim for a software invention can be drafted in several ways, including as the "process" of accomplishing a particular task through the use of a computer; or, as in *Microsoft*, as a "machine" specially programmed to accomplish the task in question. See Id at 446."

It is my contention therefore, that to reach the objectives we seek in this area, we must avoid the ambiguous term “software” and begin being more specific about what the entity at issue really is, to wit, a computer program of some specific kind, and or a computer implemented invention (CII).

A recommended framework for using more specific terms to better identify the items at issue, as well as their scope.

Accordingly, I recommend we use the following definitions.

The “software” we are talking about is a “computer program” that is defined as the "set of instructions, known as code, that directs a computer to perform specified functions or operations." And that

A claim for a computer implemented invention can be drafted in several ways, including as the "process" of accomplishing a particular task through the use of a computer; or, as a "machine" specially programmed to accomplish the task in question.

Another definitional problem that then arises is that there are many kinds of computer programs, such as

operating system program
application subprogram
subroutine
infrastructure program (GUI manager, print manager, JAVA runtime, communications manager, etc.)
client application
server application
tools (compiler, debugger, etc.)
subroutine/application program embedded in a chip /ASIC,
etc.

Each of these can perhaps be defined with some specificity, and described as having specific components. I believe it is here where the Software Technical Community can be of assistance in proposing an acceptable standard set of definitions and related minimal components for each type of computer program for use in patent application writing and claim drafting.

Another set of problems in this patenting of computer implemented invention context arises from the tension between the Patent Law rules that require a clear and concise description of what is claimed and the procedures of the Patent Bar that push Patent Attorneys/Agents to try to claim inventions as broadly (i.e. with as few restricting/defining words and structures) as possible while still obtaining a patent allowance for their client.

35 U.S.C. 112 Specification says in part, that

(a) IN GENERAL.—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 or joint inventor of carrying out the invention. And

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

On the other hand it is common for Patent Attorneys/Agents to write claims especially, with minimal elements and restrictions, and with the highest level of abstraction in defining the invention that the Patent Office will allow. This practice sometimes works to the detriment of the Inventor when the Patent Office disallows a claim as being too abstract to be patentable under section 101, as well as when courts flounder with claims allegedly describing “abstract ideas,” as in CLS and Bancorp described above.

It is in this context also that the Software Technical Community can propose to the Patent Bar, the Courts and the PTO, a set of standard definitions that can minimally define the components of the above set of computer programs/systems that the PTO and Courts would deem acceptably comply with section 112 above. Hopefully this will not intrude upon the prerogatives of the Patent Bar to advocate for their clients, but could be a reasonable set of guidelines for such claiming.

As indicated above, section 112 (a) says that the specification should describe the computer programs and processes which support the claim elements in enough detail so that a programmer skilled in the art could produce the programs without excessive difficulty (i.e like a functional specification). A question for the Technical community could be “what are the minimum requirements for such a description (functional specification)? Would it be different for the different types of programs mentioned above? For example, does a method/process for an App need to specify the infrastructure programs used (like, GPS, Oracle database, Wifi or internet), particular type of OS?

This would comply with 35 U.S.C. 112 (f), ELEMENT IN CLAIM FOR A COMBINATION.—

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.

Professor Lemley and others have proposed that elements of Computer Implemented Inventions and similar claims such as those at issue in Bilski, Bancorp, Alice, etc. be

construed as “means plus function” claims per section 112 (f) even if they do not expressly use the “means or step for” language. See Professor Lemley’s paper titled “Software Patents and the Return of Functional Claiming” at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2117302.

In this paper he concludes (page 43):

“The patent claim should accordingly be limited to that particular computer program and ones that work in the same way to achieve the same result.”

That is, the claimed invention should be limited to the particular computer program as is described in the specification. I believe that the Technical community should indicate whether they generally agree or disagree with this proposition. In any case, the possibility that this suggestion could be implemented by the Federal Circuit, should influence their suggestions for how to describe the various types of computer programs in a specification.

The comments submitted herein are my own and I am solely responsible for the contents of this response.