

**From:** Suzanne Michel  
**Sent:** Monday, April 15, 2013 8:40 PM  
**To:** SoftwareRoundtable2013  
**Subject:** Submission of Google Inc

Please find attached the submission of Google Inc. in response to the PTO's request for comments on the partnership for enhancement of quality of software-related patents, Docket No. PTO-P-2012-0052.

If possible, please send an acknowledgment of receipt of the submission.

Thank you very much,  
Suzanne Michel

--

---

Suzanne Michel | Senior Patent Counsel, Google | [stmichel@google.com](mailto:stmichel@google.com) | 202 677-5398

**Before the  
United States Patent and Trademark Office  
Alexandria, VA 22313**

In re: )  
 ) Docket No. PTO-P-2012-0052  
Request for Comments and Notice )  
of Roundtable Events for )  
Partnership for Enhancement of )  
Quality of Software-Related )  
Patents )  
 )

**COMMENTS OF GOOGLE INC.**

Daryl L. Joseffer  
KING & SPALDING LLP  
1700 Pennsylvania Avenue, NW  
Washington, DC 20006  
(202) 737-0500

Suzanne Michel  
GOOGLE INC.  
1101 New York Avenue, N.W.  
Washington, DC 20005  
(650) 253-0000

Adam M. Conrad  
KING & SPALDING LLP  
100 N Tryon Street, Suite 3900  
Charlotte, NC 28202  
(704) 503-2600

April 15, 2013

## TABLE OF CONTENTS

PART I:	INTRODUCTION .....	1
PART II:	THE PTO SHOULD APPLY SECTION 112(F) TO MORE PATENTS THAT CLAIM SOFTWARE-IMPLEMENTED INVENTIONS .....	3
	A. Section 112(f) Permits The Use Of Functional Claim Elements Only When The Specification Discloses Sufficient Structure To Limit The Claim To The Applicant’s Actual Invention .....	3
	1. As A Matter Of Policy And Precedent, Patent Law Has Never Permitted Pure Functional Claiming .....	3
	2. Congress Enacted Section 112(f) To Permit Functional Claiming Accompanied By Sufficient Disclosures. ....	4
	3. No Matter How It Is Phrased, Functional Language In Patent Claims Must Be Supported By Sufficient Structure, Material, Or Acts In The Claim Itself Or The Patent Specification.....	5
	B. Functional Claim Elements In Software Patents Should Often Be Analyzed Under Section 112(f).....	7
	1. Claims That Merely Recite A General-purpose Computer Or Software Fail To Disclose Sufficient Structure. ....	7
	2. Examples Of Claims That Do Not Provide Adequate Structure.....	9
	C. Examiners Should Follow A Systematic Approach To Determining Whether Section 112(f) Applies To Software Claims.....	11
PART III:	A DISCLOSED ALGORITHM MUST BE SUFFICIENTLY SPECIFIC TO IMPART ADEQUATE STRUCTURE TO A CLAIMED FUNCTION .....	14
	A. The Essential Characteristics Of Algorithms Are Well Known To Skilled Computer Programmers. ....	14
	B. Examples Of Patents That Do And Do Not Disclose A Sufficient Algorithm. ....	18
	1. Sufficient Disclosure.....	18
	2. Insufficient Disclosures.....	19
PART IV:	SECTION 112(F) APPLIES TO SOFTWARE METHOD CLAIMS JUST AS IT APPLIES TO OTHER CLAIMS.....	21
PART V:	EFFECTIVELY IMPLEMENTING THE SECTION 112(F) ANALYSIS.....	25
PART VI:	FUTURE DISCUSSION TOPICS.....	26

A.	Better Enforcement Of All Section 112 Doctrines.....	27
B.	Making More Software Prior Art Searchable.....	28
C.	Standardized Formats And Terminology. ....	28

APPENDIX

## **PART I: INTRODUCTION**

Clear claim boundaries and notice of a patent's scope are "essential to promote progress" and "enable[] efficient investment in innovation." *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 730-31 (2002) (citation omitted). In the software context, however, patent claims are often ambiguous, providing little in the way of public notice of their scope and ultimately impeding innovation rather than promoting it. This problem has been widely acknowledged by industry, academia, and government. Vague and overbroad software patents issued over the past 15 years have driven a litigation boom. Lawsuits brought by patent assertions entities have quadrupled since 2005 and now account for a majority of patent litigation. See RPX Corporation, *Tracking PAE Activity: A Post-script to the DOJ Review*, <http://www.rpxcorp.com/index.cfm?pageid=14&itemid=29> (last visited April 12, 2013). The vast majority of these cases—about 82%—involve software and internet patents, which are litigated eight times more often than others. See John R. Allison *et al.*, *Patent Litigation and the Internet*, 2012 Stan. Tech. L. Rev. 3; Colleen Chien & Aashish Karkhanis, *Software Patents & Functional Claiming*, [http://www.uspto.gov/patents/init\\_events/software\\_ak\\_cc\\_sw.pdf](http://www.uspto.gov/patents/init_events/software_ak_cc_sw.pdf) (Feb. 12, 2013). Indeed, the rise in litigation tracks the rise in the number of software patents. This litigation boom places a real drag on the innovation economy.

We are encouraged by the PTO's interest in improving the quality of software patents. There is no single, silver-bullet fix. But we are hopeful that the software and patent communities can work together to identify root causes, find solutions, and improve patent quality.

Parts II, III, and IV of these comments respond to "Topic 1" in the recent request for comment—establishing clear boundaries for claims that use functional language—and the specific questions posed by the PTO. See 78 Fed. Reg. 292, 294 (Jan. 3, 2013). They explain that, for over a century, Congress and the courts have allowed patent applicants flexibility to use

functional terms, but have limited the sweep of functional claims to the disclosed structure, material, and acts, as well as their equivalents. *See* 35 U.S.C. § 112(f). Although software is inherently functional, software patents have largely escaped analysis under section 112(f). The result has been broad claims that far exceed what the applicants actually invented, many of which have indiscernible scope. Those problems result in part from *not* applying traditional rules regarding functional claiming to software claims.

Going forward, the PTO should apply section 112(f) to all claim elements that are described in functional terms, not only to those that use magic words like “means for.” And it should require applicants to support and limit functional claim language with an algorithm—a specific computational procedure for achieving an input/output relationship—not with mere window dressing or empty verbiage. The PTO should also apply these requirements to system and method claims alike. Section 112(f) sets forth an important substantive limit on patentability, not one that applicants should be able to continue to evade in the application process through clever drafting.

Part V of these comments describes other, concrete actions that the PTO can take to increase clarity of software claims: formalizing section 112(f) examination procedures in guidelines and the Manual of Patent Examining Procedure; requiring applicants to identify corresponding specification support for *all* claim terms (not just functional terms); and instituting an appropriate training for examiners regarding these principles.

Finally, Part VI describes topics for future discussion to continue improving the quality of software patents, including standardizing terminology, increasing the amount of searchable prior art, and increased enforcement of all of section 112’s complementary doctrines.

**PART II: THE PTO SHOULD APPLY SECTION 112(F) TO MORE PATENTS THAT CLAIM SOFTWARE-IMPLEMENTED INVENTIONS**

An essential first step in improving the quality of software patents is to apply well-established principles regarding functional claiming to software patents.

**A. Section 112(f) Permits The Use Of Functional Claim Elements Only When The Specification Discloses Sufficient Structure To Limit The Claim To The Applicant's Actual Invention.**

**1. As A Matter Of Policy And Precedent, Patent Law Has Never Permitted Pure Functional Claiming.**

In the years prior to the 1952 Patent Act, the Supreme Court held repeatedly that patentees may not draft claims in purely functional terms. *See, e.g., Halliburton Oil Well Cementing Co. v. Walker*, 329 U.S. 1, 8-10 (1946); *Gen. Elec. Co. v. Wabash Appliance Corp.*, 304 U.S. 364, 371 (1938); *Holland Furniture Co. v. Perkins Glue Co.*, 277 U.S. 245, 256-57 (1928)). Functional language with no supporting structure lacks “definite limitation” because it can be accomplished through *any* means. *Gen. Elec. Co.*, 304 U.S. at 372. Such claims replace “structural definition” with “indeterminate adjectives,” rendering them indefinite. *Id.* at 371; *Holland Furniture*, 277 U.S. at 258 (“vague and indefinite description”). This is true even for “a combination patent embodying old elements only,” where the combination is the invention. *Halliburton*, 329 U.S. at 9; *see also id.* at 10 (requiring “clear description in combination claims” because such claims “easily lend themselves to abuse”).

The Court grounded these holdings in the fundamental *quid pro quo* underlying the patent system: inventors who disclose their inventions to the public are entitled to exclusive rights commensurate with, but no broader than, that disclosure. *E.g., J.E.M. Ag Supply, Inc. v. Pioneer Hi-Bred Int'l, Inc.*, 534 U.S. 124, 142 (2001); *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 121 (1853). When functional claims cover “the result or function of a machine,” instead of a specific machine invented by the patent applicant to perform the function, they “extend the

monopoly beyond the invention” and beyond what is described in the patent. *Holland Furniture*, 277 U.S. at 257, 258; *see also Halliburton*, 329 U.S. at 9 (rejecting claim covering “what [the invention] will do” rather than the invention itself). Put differently, purely functional claims are overbroad because they cover *all* solutions to a problem “heretofore or hereafter invented,” not just the solution or solutions a patentee actually invented and disclosed in the patent application. *Halliburton*, 329 U.S. at 12.

Such claims harm the public by deterring and penalizing further innovation. In 1946, *Halliburton* warned that, “in this age of technological development there may be many other devices beyond our present information or indeed our imagination which will perform that function and yet fit these claims. And unless frightened from the course of experimentation by broad functional claims like these, inventive genius may evolve many more devices to accomplish the same purpose.” *Id.* The rapid advancement in technological growth in recent decades has validated and underscored that warning. Then and now, the patent laws condemn “the broadness, ambiguity, and overhanging threat of . . . functional claim[s].” *Id.*

## **2. Congress Enacted Section 112(f) To Permit Functional Claiming Accompanied By Sufficient Disclosures.**

Congress responded to *Halliburton* by enacting the precursor to 35 U.S.C. § 112(f), which states that “[a]n 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.” The statute represents a middle ground between permitting unbounded functional claiming and excluding functional language altogether. Applicants may draft claim elements in terms of their functions, but those elements are limited to the specific structure, material, or acts disclosed in the patent for implementing the

function. Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2 (Stanford Public Law Working Paper No. 2117302) (2012).

The “point” of section 112(f)’s restrictions “is to avoid pure functional claiming.” *Aristocrat Techs. Austl. Pty Ltd. v. Int’l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008). By its terms, section 112(f) applies whenever a claim element (1) is expressed as a means or step for performing a specified function; and (2) does not recite structure, material, or acts to support the performance of that function. In those circumstances, the specification must cure the deficiency in the claims by disclosing a sufficient description of the structure, material, or acts. If the specification does not disclose a corresponding structure, then the claim is purely functional and therefore indefinite. *See, e.g., Blackboard, Inc. v. Desire2Learn Inc.*, 574 F.3d 1371, 1384 (Fed. Cir. 2009).

Section 112(f) thereby reconciled functional claiming with the Supreme Court’s concerns about preemption, overbreadth, and lack of clarity. Functional claim elements that are tied to a disclosed structure are commensurate with that disclosure and not overbroad. Such claims do not foreclose the entire field, nor do they raise ambiguities about what structures may or may not perform the function.

**3. No Matter How It Is Phrased, Functional Language In Patent Claims Must Be Supported By Sufficient Structure, Material, Or Acts In The Claim Itself Or The Patent Specification.**

At root, section 112(f)’s “structure, material, or acts” requirement aims to give definition to patent claims while providing clear notice of the claim boundaries to the public. *See In re Katz Interactive Call Processing Patent Litig.*, 639 F.3d 1303, 1315 (Fed. Cir. 2011) (discussing “public notice function”). “The need for clear notice is all the greater in the case of means-plus-function limitations because the claim language alone does not define the scope of protection.” *Dawn Equip. Co. v. Ky. Farms Inc.*, 140 F.3d 1009, 1023 (Fed. Cir. 1998) (Michel, J., additional

views). Thus, applying section 112(f) requires an indefiniteness analysis that “benefits the public.” *In re Avid Identification Sys., Inc.*, No. 2012-1092, 2013 WL 69102, at \*8 (Fed. Cir. Jan. 8, 2013) (Clevenger, J., dissenting).

Under section 112(f), an applicant may disclose the necessary “structure, material, or acts” in the claim itself or in the written description, but it must be in one of those two places. If a claim itself recites both a functional component and the structure that performs that function, no more is needed in this respect and section 112(f) does not apply. But if a claim recites a function without also reciting sufficient supporting structure, material, or acts, section 112(f) limits it to the structure disclosed in the specification and its equivalents. There is no permissible third category of functional claims that allows applicants to recite functional language without support in either the claims or the specification.

Thus, like other substantive limitations on patentability, applicants cannot evade section 112(f) through clever claim drafting. Section 112(f)’s ambit is not confined to claims that use magic words or catch phrases such as “means for” and “step for.” *See, e.g., Mas-Hamilton Grp. v. LaGard, Inc.*, 156 F.3d 1206, 1213-14 (Fed. Cir. 1998); *Toro Co. v. Deere & Co.*, 355 F.3d 1313, 1325 (Fed. Cir. 2004); *Mass. Inst. of Tech. v. Abacus Software*, 462 F.3d 1344, 1354 (Fed. Cir. 2006). If a claim includes terms that are ostensibly structural yet still “cover[s] every conceivable way or means to perform the function” recited, then the supposedly structural terms are not structural at all. *Mas-Hamilton*, 156 F.3d at 1214. Thus, in mechanical and electrical cases, the Federal Circuit has recognized that generic terms such as “mechanism,” “means,” “element,” and “device” typically do not connote sufficiently definite structure to limit claims in any meaningful way. *Abacus Software*, 462 F.3d at 1354.

Similarly, the standard for determining whether the recited structure, material, or acts are “sufficient” to support functional language does not depend on whether they are recited in the claim or the specification. Either way, the “structure, material, or acts” must be sufficiently specific and definite to avoid claiming *every* way of implementing the stated function (or *any* ways the applicant did not invent). That is crucial “[i]n order for the claims to serve their proper function of providing the public clear notice of the scope of the patentee’s property rights.” *Med. Instrumentation & Diagnostics Corp. v. Elekta AB*, 344 F.3d 1205, 1219 (Fed. Cir. 2003). A contrary understanding would render section 112(f) virtually meaningless and conflict with *Halliburton*’s prohibition on purely functional claims.

**B. Functional Claim Elements In Software Patents Should Often Be Analyzed Under Section 112(f).**

**1. Claims That Merely Recite A General-purpose Computer Or Software Fail To Disclose Sufficient Structure.**

When a claimed function must be performed by software, recitation of a general purpose computer or software alone is not adequate. Instead, an algorithm is a critical component of the relevant “structure.”<sup>1</sup> *Aristocrat*, 521 F.3d at 1333.

A general purpose computer can be programmed to perform very different tasks in very different ways. The same is true for the programmable *components* of a computer, such as a microprocessor, application-specific integrated circuit, or field programmable gate array. *See WMS Gaming Inc. v. Int’l Game Tech.*, 184 F.3d 1339, 1349 (Fed. Cir. 1999); *In re Aoyama*, 656 F.3d 1293, 1299 (Fed. Cir. 2011); *see also id.* at 1303 (Newman, J., dissenting). Thus, disclosing

---

<sup>1</sup> For purposes of this section, we accept the convention of the case law that calls an algorithm the required “structure” in the context of an apparatus claim containing functional claim elements. We note, however, that an algorithm will list a series of actions to be performed by a computing device and could also be called the “acts” required to support a functional claim element. This is particularly true in the context of method claims for performing a function through a series of computer-implemented steps. *See* Part IV below.

any of those components to perform a function, without also disclosing the corresponding algorithm, is akin to disclosing no structure at all. *See, e.g., Ergo Licensing, LLC v. Carefusion 303, Inc.*, 673 F.3d 1361, 1364-65 (Fed. Cir. 2012) (“control device” insufficient); *In re Katz*, 639 F.3d at 1315 (“processing means” insufficient); *Blackboard*, 574 F.3d at 1384 (“access control manager” insufficient); *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1367 (Fed. Cir. 2008) (“bank computer” insufficient); *Aristocrat*, 521 F.3d at 1332-33 (“control means” insufficient). Allowing applicants “to claim a means for performing a particular function and then to disclose only a general purpose computer as the structure designed to perform that function [would] amount[] to pure functional claiming.” *Aristocrat*, 521 F.3d at 1333.

Even though the Federal Circuit has repeatedly stated that a general purpose computer is not a structure that adequately defines a patent claim’s boundaries, claims abound that recite a computer and nothing more but yet have not been analyzed under section 112(f). Such patents include common formulations such as:

- A computer being programmed to [perform a function] or computer system providing [a function]
- A processor for [performing function] or configured to [perform function]
- Software or code for [performing function]
- Coined but essentially unlimited terms, such as a selector [for performing the function of selecting]

In the context of computer-implemented inventions, such language is analogous to generic terms that trigger section 112(f) analysis in mechanical cases, such as “means,” “mechanism,” “element,” and “device.” Under the well-settled principles of law discussed above, therefore, such claims should be analyzed under that section.

Indeed, the Federal Circuit has suggested that referring to a “processor” instead of a “computer” does not avoid section 112(f) because it does not disclose sufficient structure without also including an algorithm for performing a claimed function. *See HTC Corp. v. IPCom GmbH & Co.*, 667 F.3d 1270, 1280 (Fed. Cir. 2012). It has also recognized that the term software “commands” describes insufficient structure for similar reasons. *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1376 (Fed. Cir. 2003).

These cases and those discussed above consider whether the structure disclosed in the specification is sufficient to support a claim that is subject to section 112(f). They are equally relevant to the threshold determination whether a claim is subject to section 112(f) in the first place because, as explained above, a sufficient, limiting structure must be found in either the claims or the specification. Thus, when a patent discloses only a general-purpose computer or the like, it matters not whether that inadequate disclosure is found in the claims or the specification. Either way, the claim is purely functional and thus unpatentable.

Indeed, the Patent Trial and Appeal Board has issued several recent decisions entering new grounds of rejection under section 112(f) for claims that include only a “processor” as structure in the claims. *See Ex parte Smith*, Appeal 2012-007631, slip op. at 3 (PTAB Mar. 14, 2013) (“processor . . . programmed to”); *Ex parte Lakkala*, Appeal 2011-001526, slip op. at 4 (PTAB Mar. 13, 2013) (“processor . . . configured . . . to”); *Ex parte Erol*, Appeal 2011-001143, slip op. at 9 (PTAB Mar. 13, 2013) (“processor adapted to”). This is a welcome step that will help bring PTO examination practice in line with the statute and controlling case law.

## **2. Examples Of Claims That Do Not Provide Adequate Structure.**

Unfortunately, many issued claims that should have been analyzed and rejected under section 112(f) were not. Consider the following illustrative claims, which are reproduced in full in the appendix. U.S. Patent No. 5,715,314, which the Federal Circuit recently invalidated on

other grounds, relates to electronic commerce technology. *See Sovereign Software LLC v. Newegg*, 705 F.3d 1333 (Fed. Cir. 2013). Claim 34 of the '314 patent recites the limitation “said buyer computer being programmed to receive a plurality of requests from a user to add a plurality of respective products to a shopping cart.” That claim element essentially consists of a general purpose computer (called a “buyer computer”) that performs a function of receiving requests as a result of certain programming. The reference to a computer by itself is insufficient structure, as discussed above, and the claim does not include even a hint of an algorithm.

Similarly, U.S. Patent No. 6,466,862 relates to a system for providing traffic information. Claim 1, which was later cancelled during reexamination, recites the limitation “said computer system, in response to a request for traffic information from one of said mobile user stations, providing in response thereto to said one of said mobile user stations traffic information representative of said signals transmitted by said traffic monitors.” This, too, is a functional claim. The phrase “providing . . . traffic information” is a function that must be performed by software on the claimed “computer system.” But the claims do not disclose, or limit the computer system to, any particular structure.

The PTO has also issued patent claims that refer only generically to software or code, without more. U.S. Patent No. 6,981,007, for example, generally relates to backing up data over the internet. Claim 1 includes two “software” limitations: “software executing on said central computer for receiving a data backup request from said client computer”; and “software executing on said central computer for transmitting said data backup to said client computer for onsite backup of internet-based data on said client computer.” Under *Altiris*, however, the term “software” alone is not sufficient structure to support the claimed “receiving” and “transmitting” functions. *See Altiris*, 318 F.3d at 1376.

Finally, and perhaps most troubling, there is a growing trend in which applicants simply coin new terms that add nothing in the way of disclosure or limitation. By way of example, U.S. Patent No. 6,327,579 relates to online computer services, such as renting software over the internet. Claim 13 recites the limitation “a selector that allows the customer to select a software application for rental from said software rental service.” The ostensibly structural term “selector” does nothing more than restate what follows: the function of allowing the customer to select software. Thus, the coined term serves, at most, to obscure the lack of a disclosed structure. The claim goes on to use this approach repeatedly and flagrantly, reciting “a validator that validates” and a “determiner that determines,” among others. Because these phrases just call for a computer or software with means for allowing, validating, or determining, they are functional limitations subject to section 112(f).

This kind of careful, case-by-case analysis of functional claim language would not require special rules for software patents. It simply reflects a straightforward application of the statute and binding Supreme Court precedent, as discussed above. These claims cannot be exempted from section 112(f) under leading Federal Circuit cases (such as *Aristocrat* and its progeny) that analyze software claims under the statute. Indeed, allowing software patent applicants to draft around the statute by reciting a computer, a generic computer component, or making up terms (such as “selector”) that parrot the claimed function would effectively insulate software patents from the prohibition on pure functional claiming—thus treating software patents differently from other patents.

**C. Examiners Should Follow A Systematic Approach To Determining Whether Section 112(f) Applies To Software Claims.**

The PTO can and should ensure that software patents have clear boundaries and appropriate scope by determining in every case whether section 112(f) applies and creating a

clear record of the analysis. We suggest the following guidelines for examiners in determining whether to apply section 112(f) to a software claim.

**Reject Presumptions.** The MPEP currently embraces a rebuttable presumption that section 112(f) does not apply in the absence of the phrases “means for” or “step for” and emphasizes that the “presumption is a strong one that is not readily overcome.” MPEP § 2181. For the reasons explained above, we recommend a more flexible and less arbitrary approach.

It is well settled that the PTO must apply section 112(f) when the statute requires it. *See In re Donaldson*, 16 F.3d 1189 (Fed. Cir. 1994) (en banc). Artificial presumptions and rigid rules cannot defeat that legal requirement. Indeed, presumptions are especially inappropriate in the context of prosecution, where one goal is to create a clear record of claim scope and applicants have the ability to clarify the claims in response. The PTO “has a duty to guard the public against patents of ambiguous and vague scope.” *Ex parte Miyazaki*, No. 2007-3300, slip op. at 12 (BPAI Nov. 19, 2008). Applicants are free to avoid section 112(f) by explaining why language is not functional, by removing functional language, or by adding sufficient structure to the claims to prevent application of section 112(f). The examination process is designed to clarify claim scope, but artificial presumptions tend to obscure it. The PTO’s bright-line presumption makes it easy for patent applicants to evade section 112(f) by not using magic words such as “means for,” even when a claim is clearly functional, as in the above examples.

To be sure, MPEP section 2181 has identified a non-exhaustive list of non-structural words that simply substitute for the phrase “means for.” *See* MPEP § 2181 (listing “mechanism,” “module,” “device,” “unit,” “component,” “element,” “member,” “apparatus,” “machine,” and “system”). But that simply confirms the need for flexibility and the inappropriateness of a presumption based on the catch phrase “means for.” Once one recognizes

that other phrases may also trigger section 112(f), and that patentees can always respond to any list of magic words by using some other formulation, the PTO's presumption becomes untenable.

Moreover, the terms currently listed in the MPEP are far more relevant to machines than software, which has its own, evolving vocabulary. As the examples above show, there is a growing number of ways to refer to generic structures in computer hardware and software. Phrases such as a "computer programmed to [perform a function]," a processor for [performing a function]," or "software for [performing a function]," are more natural ways to describe computer-implemented inventions in functional terms. Strict adherence to the "means" paradigm for software patents is bad policy that is flatly inconsistent with section 112(f).

**Make a Clear Record.** Examiners should inform applicants which elements are subject to section 112(f)'s requirements. There should be an open dialogue on the subject that is fully incorporated into the written record. As the Federal Circuit has stressed, "it is highly desirable" for patent examiners to raise such discussions "so that the patent can be amended during prosecution rather than attempting to resolve the ambiguity in litigation." *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1255 (Fed. Cir. 2008). Thus, examiners should require applicants to particularly point out supporting structure in the specification for claim elements that are subject to section 112(f). We support the PTO's proposal to require all applicants to do this. As discussed below, it may be appropriate to reject claims as indefinite either before or after the applicant responds. In other cases, it may be appropriate to offer a tentative claim construction regarding application of section 112(f) and the structure that the Examiner believes supports that construction.

**Reject Intent Evidence.** Examiners should not base a section 112(f) determination on the inventor's intent. The question whether a claim recites sufficient structure is governed by the

statute and the applicant's disclosure to the public, not the inventor's subjective intent. For example, an applicant could not draft a claim using the traditional "means for" formulation and then evade section 112(f) by asserting an intent to have that phrase disclose sufficient structure.

In addition, experience has shown that applicants would likely decline section 112(f) treatment if given a chance to do so. Just in the last decade, the number of patent applications that use traditional "means for" language have dropped by well over half. See "Means Plus Function Claiming," PatentlyO (Jan. 14, 2013), *available at* <http://www.patentlyo.com/patent/2013/01/means-plus-function-claiming.html>. That does not mean that applicants have abandoned functional claiming, just that presumptions and magic-word tests are easily manipulated. The examples discussed above make this clear. Giving patent applicants the opportunity to "deem" non-structural language to be structural would only increase uncertainty, detract from clarity, and even potentially put otherwise deserving patents at risk of invalidation. Thus, we do not view as helpful the PTO's suggestion to ask applicants to check a box indicating whether they intend for section 112(f) to apply, *i.e.*, to ask whether the patentee would like to opt out of this substantive requirement for patentability.

### **PART III: A DISCLOSED ALGORITHM MUST BE SUFFICIENTLY SPECIFIC TO IMPART ADEQUATE STRUCTURE TO A CLAIMED FUNCTION**

Determining whether section 112(f) applies is just the first step. The second step is determining whether the specification discloses an algorithm to support a claimed software function and whether that algorithm is sufficiently definite to satisfy the statute.

#### **A. The Essential Characteristics Of Algorithms Are Well Known To Skilled Computer Programmers.**

As noted above, an algorithm is a critical component of the "structure" for computer-implemented inventions. *Aristocrat*, 521 F.3d at 1333. And "[t]he 'structure, material or acts' that must support a claim in functional language must be more than mere window-dressing."

Lemley, at 43. Although section 112(f) does not use the term “algorithm,” the defining attributes of algorithms are well known to those of skill in the art. Indeed, an algorithm is the very essence of a computer-implemented invention; it is how the invention works.

As one leading programming textbook has explained, an “algorithm describes a specific computational procedure for achieving [an] input/output relationship.” Thomas H. Cormen et al., *Introduction to Algorithms* 5 (2d ed. 2001). Significantly, that means that an algorithm is *not* synonymous with “[t]he statement of the problem,” which simply “specifies in general terms the desired input/output relationship.” *Id.* This fundamental definition provides a sound starting point for testing a patent application’s disclosure.

Indeed, the root distinction between stating a relationship in “general terms” and detailing “a specific computational procedure” lies at the heart of much of the Federal Circuit’s recent case law in this area. Within the past year, the court validated the PTO’s view that “a high level process flow” is not a sufficient algorithm. *Aoyama*, 656 F.3d at 1298. Similarly, a disclosure that simply “describes the function to be performed” or “describes an outcome” lacks the necessary structure. *Aristocrat*, 521 F.3d at 1334. At a minimum, a disclosed algorithm must include “some detail about the means to accomplish the function,” which is, by definition, the computational procedure performed by software. *Finisar Corp. v. DirecTV Group, Inc.*, 523 F.3d 1323, 1340-41 (Fed. Cir. 2008). It is not enough, however, to require that an algorithm be “specific” rather than “general.” Such a relative test would lend itself to ready evasion in the absence of concrete guideposts.

Another highly regarded text authored by Turing Award winner, Donald Knuth, shows what is required. *See* Donald E. Knuth, *The Art of Computer Programming* 4 (3d ed. 1997). It explains that algorithms share important features, several of which are relevant here:

- *First*, “An algorithm must always terminate after a finite number of steps.”
- *Second*, “Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case.”
- *Third*, “An algorithm has zero or more inputs: quantities that are given to it initially before the algorithm begins, or dynamically as the algorithm runs.”
- *Fourth*, “An algorithm has one or more outputs: quantities that have a specified relation to the inputs.”

*Id.* at 4-6. Put another way, an algorithm must identify the inputs, the outputs, and—critically—enough detail to allow someone to take the actions necessary to generate the outputs from the inputs. A purported algorithm that fails to satisfy these guideposts does little more than restate general functions and therefore lacks the implementation details required by section 112(f).

In some cases, applicants may satisfy this requirement by pointing to well-known algorithms in the prior art. Applicants can efficiently identify existing algorithms (or other structures) without adding unnecessary detail to the patent specification.

Some may object that an abstract, high-level flow chart is sometimes sufficient to allow skilled artisans to implement a claimed function. But that is a question of *enablement* under section 112(a), not a question of *definiteness* under section 112(f). “Enablement of a device requires only the disclosure of sufficient information so that a person of ordinary skill in the art could make and use the device. A [section 112(f)] disclosure, however, serves the very different purpose of limiting the scope of the claim to the particular structure disclosed, together with equivalents.” *Aristocrat*, 521 F.3d at 1336. Indeed, “it is well established that proving that a person of ordinary skill *could* devise some method to perform the function is not the proper

inquiry as to definiteness . . . .” *Function Media, L.L.C. v. Google Inc.*, 708 F.3d 1310, 1319 (Fed. Cir. 2013) (emphasis in original); *see also Blackboard*, 574 F.3d at 1385.

The purpose of section 112(f)—and thus the purpose of requiring an algorithm to support functional software claims—is to ensure “adequate defining structure to render the *bounds of the claim* understandable to one of ordinary skill in the art.” *AllVoice Computing PLC v. Nuance Commc’ns, Inc.*, 504 F.3d 1236, 1245 (Fed. Cir. 2007) (emphasis added); *accord ePlus, Inc. v. Lawson Software, Inc.*, 700 F.3d 509, 519 (Fed. Cir. 2012) (“The indefiniteness inquiry is concerned with whether the bounds of the invention are sufficiently demarcated, not with whether one of ordinary skill in the art may find a way to practice the invention.”). A patent that purports to disclose an algorithm but fails to specify any inputs, outputs, or the computational steps to connect them does not provide any boundaries at all, let alone clear ones.

By conflating section 112(f) with enablement, evaluating the disclosed algorithm in light of the undisclosed knowledge of a person of ordinary skill would allow pure functional claiming, contrary to Congress’s clear intent. “That ordinarily skilled artisans could carry out the recited function in a variety of ways is precisely why claims written in ‘means-plus-function’ form must disclose the particular structure that is used to perform the recited function.” *Blackboard*, 574 F.3d at 1385. “[A] patentee cannot avoid providing specificity as to structure simply because someone of ordinary skill in the art would be able to devise a means to perform the claimed function. To allow that form of claiming under section [112(f)] would allow the patentee to claim all possible means of achieving a function.” *Function Media*, 708 F.3d at 1319 (quoting *Blackboard*, 574 F.3d at 1385).

Thus, a disclosure that relies on background knowledge in the art, without pointing to or incorporating particular known algorithms, is by definition inadequate. *See ePlus*, 700 F.3d at

519 (without disclosure of a particular algorithm, the specification did not cabin the functional language and the patentee has “claimed everything that [performs the recited function] under the sun”). A sufficient disclosure must describe *how* software accomplishes the recited function to avoid preempting every possible way of performing that function. *See Function Media*, 708 F.3d at 1318-19 (rejecting flow chart as sufficient where it did not describe how software performed recited function). A contrary approach would expand the claim’s boundaries rather than define them, contrary to the principle that “a patentee cannot obtain greater coverage by failing to describe his invention than by describing it as the statute commands.” *Halliburton*, 329 U.S. at 13.

**B. Examples Of Patents That Do And Do Not Disclose A Sufficient Algorithm.**

**1. Sufficient Disclosure.**

U.S. Patent No. 6,061,703 is a good example of a patent that recites functional claim language supported by an algorithm disclosed in the specification. Claim 16 of the ’703 patent is written in traditional “means for” format; thus, there is no question that this claim requires application of section 112(f). The specification also discloses clear, detailed algorithms to support the claimed functions, including the following examples:

- Claim 16 recites a “means for generating a pseudorandom number as a function of said secret value.” The supporting algorithm includes the F3 and XOR 248 components of Figure 2. As the specification explains, “function F3 concatenates (240) T and S in that order and passes the 192-bit concatenation result through a one-way hash function 242 (such as MDC-4) to generate the 128-bit hash value 238.” Col. 7:15-20. The left 64 bits and the right 64 bits of the hash value are then “exclusive-ORed (248) . . . to produce an output comprising a 64-bit pseudorandom number (PRN) 250.” Col. 7:20-22.

- Claim 16 also recites “means for updating said secret value with a first updated secret value in response to a first normal mode stimulus in said normal mode of operation.” The supporting algorithm includes feedback function F1 in Figure 2. The specification states that “feedback function F1 concatenates (214) S and T in that order and passes the 192-bit concatenation result through a one-way hash function 216 (such as the MDC-4 function described below) to generate a 128-bit hash value 218.” Col. 6:58-62. Gate 220 “passes the hash value 218 back to S register 208.” Col. 6:64-65.

These examples satisfy the requirements of an algorithm by disclosing the inputs (S and T), the outputs (pseudorandom numbers and updated secret values), and the specific concatenation, XOR, and hash operations necessary to generate the outputs from the inputs. The specification makes clear that the MDC-4 hash function is well known in the art and also describes it in the specification. *See* Col. 10:64-66.

If the '703 patent had instead included a high-level flow chart while excluding any of these features, it would *not* have satisfied section 112(f). Excluding any of these computational procedures would remove the structure that tells a person of skill in the art *how* the claimed function is performed. And it would thereby preempt every way to perform the claimed function.

## **2. Insufficient Disclosures.**

The four examples of functional claims discussed above in Part II are also examples of patents with insufficient disclosures. U.S. Patent No. 6,981,007 is an extreme instance of a software patent with functional claims and no supporting algorithm. The two “software” limitations lack any meaningful corresponding support in the specification. The Figures consist of routine depictions of client and server computers connected over the internet, and the

“Detailed Description” consists of just four paragraphs of text. Instead of disclosing an algorithm, the specification simply parrots the claimed function, referring to computers that “execute[] software” to perform various functions. *See, e.g.*, Col. 3:1-5. The patent does not even identify existing software programs in lieu of an algorithm. Thus, this disclosure is little more than a reference to “appropriate programming,” which “imposes no limitation whatever, as any general purpose computer must be programmed.” *Aristocrat*, 521 F.3d at 1334.

U.S. Patent No. 5,715,314 is deficient for essentially the same reason. The specification does not expand on the claim’s functional language—“programmed to receive a plurality of requests from a user to add a plurality of respective products to a shopping cart”—in any meaningful way. In the Summary of the Invention, the specification simply repeats that “[t]he buyer computer is programmed to receive a user request for purchasing a product” or “is programmed to receive a plurality of requests from a user to add a plurality of respective products to a shopping cart in the shopping cart database.” Col. 1:55-57; 2:25-28. The “Detailed Description” is, if anything, even less detailed and forthcoming. *See, e.g.*, Col. 5:27-28 (user “requests a product”). By simply “restat[ing] the function associated with the means-plus-function limitation,” the ’314 patent’s specification does nothing to limit the claim or illuminate its boundaries. *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1317 (Fed. Cir. 2012).

Claim 1 of U.S. Patent No. 6,466,862 similarly lacks an adequate algorithm to support the claimed “providing . . . traffic information” function. Figures 9 and 10 provide only a “high level process flow.” *Aoyama*, 656 F.3d at 1298. The specification actually emphasizes the absence of any limiting structure by stating that the computer “may manipulate the traffic information *in some manner*, as necessary, so as to provide average speeds or other statistical data.” ’862 patent, Col. 8:55-58 (emphasis added). In addition, the computer “may” transmit

average speeds, and “may” send data corresponding to some monitors but not others. Col. 9:19-27. At best, this language “describes an outcome” of performing a function. *Aristocrat*, 521 F.3d at 1334. The specification includes no algorithm for distinguishing between these outcomes or for implementing them, and it fails all of the basic requirements of defining the inputs, outputs, and computational procedures that typify algorithms.

Finally, claim 13 of U.S. Patent No. 6,327,579 recites selecting, validating, and determining functions that have no corresponding algorithm in the specification. Figures 8A and 21A, for example, repeat the “validate” function as part of a flowchart with no additional detail. And despite its length, the specification adds no further detail, stating only that “[c]ustomer commands and menu requests are received and validated,” Col. 24:48-49, and that a “Host Security Program” operates “to validate access requests for virtual disks,” Col. 28:48-49.

Each of these examples “vividly illustrates the vice of a description in terms of function.” *Gen. Elec.*, 304 U.S. at 371. These patents attempt to gloss over the lack of an actual, innovative invention by vaguely reciting a computer or software for accomplishing a result, without disclosing any corresponding structure or algorithm. *Blackboard*, 574 F.3d at 1385. They are purely functional and pose the very overbreadth and preemption problems section 112(f) was designed to avoid.

#### **PART IV: SECTION 112(F) APPLIES TO SOFTWARE METHOD CLAIMS JUST AS IT APPLIES TO OTHER CLAIMS**

Although the preceding sections generally concern system claims, the same principles apply to method claims as well. Section 112(f) requires that claims “expressed as a . . . step for performing a specified function without the recital of . . . acts in support thereof . . . shall be construed to cover the corresponding . . . acts described in the specification and equivalents thereof.” 35 U.S.C. § 112(f). Thus, whereas section 112(f) requires that functional systems

claims be supported by recital of structure, as discussed above, it requires that functional method claims be supported by description of “acts.” *O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 1582 (Fed. Cir. 1997). That dichotomy reflects the basic difference between system and method claims: the former cover a structure while the latter cover a series of steps or acts to be performed.

The Federal Circuit has recognized that, when analyzing whether a method of using mechanical devices falls within section 112(f), it can be difficult to distinguish between a step for performing a function and acts performed in support of that step. Acts and functions both “are often stated using verbs ending in ‘ing.’” *See Seal-Flex, Inc. v. Athletic Track & Court Constr.*, 172 F.3d 836, 849 (Rader, J., concurring). Chief Judge Rader provided a helpful distinction, however:

In general terms, the “underlying function” of a method claim element corresponds to *what* that element ultimately accomplishes in relationship to what other elements of the claim and the claim as a whole accomplish. “Acts,” on the other hand, correspond to *how* the function is accomplished.

*Id.* at 849-850 (emphasis in original). If a method claim containing functional elements does not also include “acts” corresponding to *how* the function is accomplished, it falls within the framework of section 112(f). *See Masco Corp. v. United States*, 303 F.3d 1316, 1327-28 (Fed. Cir. 2002) (holding that section 112(f) did not apply because disputed claim language itself included the “act” of *how* function was performed).

In the software context, therefore, a functional step in a method claim must include a corresponding algorithm. *See Function Media*, 708 F.3d at 1318-19 (rejecting flow chart as sufficient algorithm to support functional claim element because it did not describe how the

function was performed). Because algorithms define *how* software performs a recited function, they specify the acts necessary to perform the function of generating the outputs from the inputs. Knuth, *The Art of Computer Programming* 4. In that sense, system and method claims parallel each other: an algorithm provides the relevant *structure* for the functional software elements of systems claims and the relevant *acts* for the functional software steps of method claims. Thus, *Aristocrat*'s key insight—that functions performed by software on a general purpose computing device amount to pure functional claiming unless limited by an algorithm—applies with equal force to method claims. See *Aristocrat*, 521 F.3d at 1334. When method claims recite a functional step performed by software, the 112(f) analysis must ask: (1) does the claim include a sufficient algorithm constituting “acts” to support the functional step; and (2) if not, is such an algorithm present in the specification?

Without this restriction, method claims would open a giant loophole in section 112(f), allowing ready evasion of the statute's and *Halliburton*'s prohibition on pure functional claiming. Because most computer-implemented inventions could be described and claimed as both a system and a method, patent applicants commonly include corresponding system and method claims in the same patent. As the Federal Circuit has recognized in other contexts, “the form of the claims should not trump basic issues of patentability.” *Bancorp Servs., L.L.C. v. Sun Life Assurance Co. of Can.*, 687 F.3d 1266, 1277 (Fed. Cir. 2012). Ignoring the important role of the algorithm in method claims while recognizing it in system claims would elevate form over substance while ceding important questions of disclosure and patentability to the whims of the “draftsman's art.” *Parker v. Flook*, 437 U.S. 584, 593 (1978).

For example, system claim 16 of the U.S. Patent No. 6,061,703 is virtually identical to method claim 1, which essentially strips out the phrase “means for” from each limitation of claim 16 while leaving the claimed functions. For example:

- “means for storing a secret value” in the system claim, Col. 16:28, becomes “storing a secret value” in the method claim, Col. 13:62; and
- “means for generating a pseudorandom number as a function of said secret value” in the system claim, Col. 16:30-31, becomes “generating a pseudorandom number as a function of said secret value” in the method claim, Col. 13:63-64.

In both cases, the change is purely one of form—a drafter’s choice, rather than a change in claim scope. Given that claim 16 is indisputably a means-plus-function claim limited to the disclosed algorithm, it would make no sense to construe claim 1 to go beyond that disclosure. Doing so would amount to pure functional claiming in one format that was prohibited in the other.

The same is true for patent claims that do not use the traditional “means for” language. In U.S. Patent No. 6,327,579, discussed above, many limitations in method claim 11 and system claim 13 mirror each other:

- compare “a validator that validates the customer’s authority to access application services provided by said software rental service” in claim 13, Col. 74:1-3, with “validating the customer’s authority to access application services provided by said software rental service” in claim 11, Col. 72:39-40; and
- compare “an execution determiner that determines if a software component of said selected software application should be executed at the customer’s computer or remotely thereto” in claim 13, Col. 74:7-10, with “determining if a software

component of said software application selected in step (d) should be executed in the customer's computer processor" in claim 11, Col. 72:43-45.

As these examples confirm, there is no principled reason to treat functional language in method claims differently than the very same language in system claims. In each of these patents, the method claim elements are *steps* for performing a function that correspond identically (or virtually so) to the means for performing the same function in the system claim. That the method steps describe functions is therefore indisputable.

The Federal Circuit in *Masco* declined to apply section 112(f) to a method claim because it found acts within the claim itself—not because it found the language of the method claim not to be functional. In the examples from the '703 and '579 patents discussed above, however, it is clear that no claim language describes the algorithm that software must use to perform the functions. As a result, these claims must be supported by an algorithm in the specification to avoid pure functional claiming and indefiniteness.

#### **PART V: EFFECTIVELY IMPLEMENTING THE SECTION 112(F) ANALYSIS**

The analysis discussed above should be fully memorialized through modification of the section 112 examination guidelines and the Manual of Patent Examining Procedure. For example, the PTO should amend MPEP § 2181 to make clear that application of section 112(f) does not turn on the use of magic words or catch phrases. The use of "means" may well indicate that section 112(f) applies, but especially in the context of software claims, it makes little sense to emphasize that term over all others. The examination guidelines should make clear that the test for sufficient structure is the same whether the purported structure appears in the claims or in the specification.

In addition, the PTO's Request for Comments on Preparation of Patent Applications asks for comment on advantages and disadvantages of requiring applicants to identify "corresponding

support in the specification” for each element for all claims. 78 Fed. Reg. 2960, 2960-61 (Jan. 15, 2013). We believe the PTO should implement this requirement and formalize guidelines to that effect. This standardized approach will bring uniformity to examination procedures while helping examiners analyze all of the relevant section 112 requirements.

The requirement will be particularly advantageous for claims that fall under section 112(f) because those claims must be construed according to the specification support. This approach would give applicants the opportunity to clarify their claims and focus the examination process by identifying the structure, material, or acts that correspond to a claimed function. Resolving these ambiguities during examination may prevent patentees from later asserting a far broader claim scope during litigation. As discussed above, however, requiring applicants to identify support in the specification does not mean that application of section 112(f) turns on an applicant’s intent. Whether the statute applies is an objective inquiry into the patent’s scope and whether the claims serve the public notice function required by the Patent Act.

In conjunction with modifying its guidelines and the MPEP, the PTO should also adopt corresponding training for examiners. Section 112(f) has been poorly enforced during prosecution of software patents—an afterthought left to follow-on litigation for meaningful analysis. It is essential to pair any new substantive guidance with a training regime that emphasizes section 112(f)’s importance and application, especially for examiners in art units where claims may include functional elements performed by software.

## **PART VI: FUTURE DISCUSSION TOPICS**

In topic 2 of the request for comment, the PTO requested suggestions for future discussion topics for the Software Partnership. 78 Fed. Reg. at 294. We are encouraged by the PTO’s willingness to have an ongoing dialogue and suggest further discussion in the following

areas. We provide three suggestions, but these are certainly not exhaustive of the issues related to software patent quality.

**A. Better Enforcement Of All Section 112 Doctrines.**

The requirements of section 112 (definiteness, written description, and enablement) are rarely enforced in the software area. But these requirements serve the important purpose of providing public notice and clear claim boundaries. Indeed, all of these requirements work in tandem: requiring disclosure of a sufficient algorithm under section 112(f), for example, also helps ensure that the applicant had possession of the invention and complied with the written description requirement. Insufficient attention to these requirements has resulted in the issuance of many vague, overbroad software and Internet patents that fuel litigation and have a chilling effect on innovation. Future discussions should explore how better to apply all section 112 requirements to software patents during examination.

This discussion would be greatly aided by making more information publicly available about the trends in enforcement by examiners. As part of its Quality Metric program, the PTO's Office of Patent Quality Assurance already collects data on the application of section 112 during examination, including whether a rejection under section 112 should have been made but was not. *See* U.S. Patent Trademark Office, *Adoption of Metrics for the Enhancement of Patent Quality*, available at [http://www.uspto.gov/patents/init\\_events/qual\\_comp\\_metric.pdf](http://www.uspto.gov/patents/init_events/qual_comp_metric.pdf). However, this data is only available in the aggregate values used to calculate the composite quality score. The PTO should consider making public all of the underlying data to the composite score so that it can be observed if section 112 is being consistently and properly enforced, and to compare its application to software patents with its application in other technology areas.

**B. Making More Software Prior Art Searchable.**

A significant amount of software-related prior art does not exist in common databases of issued patents and published academic literature. For instance, it may be embedded in computer code or detailed in non-digitized manuals. Cataloging and making this prior art searchable by examiners and the public would help curtail the issuance of invalid patent claims. It could also ease the burdens on examiners with limited time and resources. Future discussions could explore how to make more prior art available to examiners and the public.

**C. Standardized Formats And Terminology.**

The software field lacks the standardized terminology of the chemical field. For this reason, and because applicants benefit from ambiguity and the ability to stretch claims in later infringement proceedings, they often use vague or ill-defined terminology in software patent claims. When the same concept is described in different applications using different terminology, prior art searches (for examiners) and clearance searches (for innovators) are less effective. Moreover, the uncertain scope resulting from ill-defined claim terms fuels litigation. An important topic for future discussion is what steps the Patent Office and industry can take to help standardize and clarify software terminology in patents. Patent applicants have it within their power to clarify terms during prosecution, and they should do so, rather than leaving claim scope to costly litigation after the fact.

A related question is whether a requirement to use pseudo-code notation systems to describe software inventions would improve the disclosure value of the specification and the clarity of the claim boundaries. The current lack of a standard format for pseudo-code undermines the value of such a proposal. Without generally accepted criteria for describing software through pseudo-code, descriptions could vary greatly and applicants who wish to provide only sketchy information could continue to do so. A topic for future discussion could be

April 15, 2013

Page 29

whether a standard notation format could be created and whether requiring its use would be beneficial for the disclosure of software inventions.

# **APPENDIX**

Full Text Of Claims Cited In Comments

**U.S. Patent No. 5,715,314 – Claim 34**

34. A network-based sales system, comprising:  
at least one buyer computer for operation by a user desiring to buy products;  
at least one shopping cart computer; and  
a shopping cart database connected to said shopping cart computer;  
said buyer computer and said shopping cart computer being interconnected by a computer network;

said buyer computer being programmed to receive a plurality of requests from a user to add a plurality of respective products to a shopping cart in said shopping cart database, and, in response to said requests to add said products, to send a plurality of respective shopping cart messages to said shopping cart computer each of which comprises a product identifier identifying one of said plurality of products;

said shopping cart computer being programmed to receive said plurality of shopping cart messages, to modify said shopping cart in said shopping cart database to reflect said plurality of requests to add said plurality of products to said shopping cart, and to cause a payment message associated with said shopping cart to be created; and

said buyer computer being programmed to receive a request from said user to purchase said plurality of products added to said shopping cart and to cause said payment message to be activated to initiate a payment transaction for said plurality of products added to said shopping cart;

said shopping cart being a stored representation of a collection of products, said shopping cart database being a database of stored representations of collections of products, and said shopping cart computer being a computer that modifies said stored representations of collections of products in said database.

## U.S. Patent No. 6,061,703 – Claims 1 and 16

1. A method for pseudorandomly generating numbers capable of use in a cryptographic system, comprising the steps of:

storing a secret value;

generating a pseudorandom number as a function of said secret value;

specifying one of a plurality of modes of operation including a normal mode of operation and a test mode of operation;

updating said secret value with a first updated secret value in response to a first normal mode stimulus in said normal mode of operation, said updating step being inhibited in said test mode of operation;

updating said secret value with a second updated secret value in response to a second normal mode stimulus in said normal mode of operation, said latter updating step being inhibited in said test mode of operation;

updating said secret value with said first updated secret value in response to a first test mode stimulus in said test mode of operation; and

updating said secret value with said second updated secret value in response to a second test mode stimulus in said test mode of operation.

16. A pseudorandom number generator capable of use in a cryptographic system, comprising:

means for storing a secret value;

means for generating a pseudorandom number as a function of said secret value;

means for specifying one of a plurality of modes of operation including a normal mode of operation and a test mode of operation;

means for updating said secret value with a first updated secret value in response to a first normal mode stimulus in said normal mode of operation, said updating means being inhibited in said test mode of operation;

means for updating said secret value with a second updated secret value in response to a second normal mode stimulus in said normal mode of operation, said latter updating means being inhibited in said test mode of operation;

means for updating said secret value with said first updated secret value in response to a first test mode stimulus in said test mode of operation; and

means for updating said secret value with said second updated secret value in response to a second test mode stimulus in said test mode of operation.

**U.S. Patent No. 6,327,579 – Claims 11 and 13**

11. A method of providing Internet software application rental services to customer computer users comprising:

(a) establishing a customer relationship between a customer and an application service provider providing a software rental service,

(b) storing in a data format at said software rental service at least one of the following items associated with the customer:

(1) a user identifier,

(2) a password, and

(3) a credit card identifier;

(c) establishing an Internet connection between at least one computer associated with the customer and the application service provider;

(d) validating the customer's authority to access application services provided by said software rental service;

(e) selecting a software application for rental from said software rental service;

(f) determining if a software component of said software application selected in step (d) should be executed in the customer's computer processor;

(g) executing at least one software component of said software application selected in step (e) on a computer of said software rental service, said execution conditioned at least in part by said determining step (f); and

(h) transmitting for execution, in the customer's computer, at least one software component of said software application selected in step (e) from said software rental service's computer to said customer's computer, said transmitting conditioned at least in part by said determining step (f).

13. In a system for providing Internet software application rental services to customer computer users, a software rental service provider comprising:

a storage device that stores in a data format at least one of the following items associated with a customer:

(1) a user identifier,

(2) a password,

(3) a name, and

(4) a credit card number;

a communications facility that establishes an Internet connection between at least one computer associated with the customer and the software rental service provider;

a validator that validates the customer's authority to access application services provided by said software rental service;

a selector that allows the customer to select a software application for rental from said software rental service;

an execution determiner that determines if a software component of said selected software application should be executed at the customer's computer or remotely thereto;

a processor coupled to the execution determiner that conditionally executes at least one software component of said selected software application;

the communications facility including a component transmitter that conditionally transmits at least one software component of said selected software application to said customer's computer for execution.

**U.S. Patent No. 6,466,862 – Claim 1**

1. A system for providing traffic information to a plurality of mobile users connected to a network, comprising:

(a) a plurality of traffic monitors, each said traffic monitor comprising at least a detector and a transmitter, said detector providing a signal including data representative of vehicular movement and said transmitter transmitting said signals;

(b) a receiver, remotely located from said transmitter, that receives said signals transmitted by said traffic monitors; and

(c) a computer system interconnected with said receiver and said network;

(d) a mobile user station connected to a global positioning system receiver, a display, and a communicating device; and

(e) said computer system, in response to a request for traffic information from one of said mobile user stations, providing in response thereto to said one of said mobile user stations traffic information representative of said signals transmitted by said traffic monitors;

(f) wherein said traffic information transmitted by said computer system is displayed graphically on said display; and

(g) wherein said computer system has a map database, and said computer system, in response to said request for information, transmits map information representative of a portion of said map database, and said map information representative of said map database is displayed graphically together with said traffic information.

**U.S. Patent No. 6,981,007 – Claim 1**

1. A system for onsite backup of internet-based data comprising:
  - a central computer;
  - a client computer;
  - a communications link between said central computer and the Internet;
  - a communications link between said client computer and the Internet;
  - at least one database containing a plurality of data records accessible by said central computer, each data record containing a client identification number;
  - software executing on said central computer for receiving a data backup request from said client computer;
  - software executing on said central computer for transmitting said data backup to said client computer for onsite backup of internet-based data on said client computer.