Comment to PTO–P–2012–0052/Request for Comments and Notice of Roundtable Events for Partnership for Enhancement of Quality of Software-Related Patents[1]

Colleen V. Chien[2] and Aashish R. Karkhanis[3]
Santa Clara University School of Law

Abstract

On Feb 12, 2013, the PTO held a roundtable about software patents. Software patents have received a lot of attention and we don't believe it is undue: software patents are behind a disproportionate share of patent litigations -- more specifically, over half (55%) of all patent defendants and 82% of NPE ("non-practicing entity") defendants are there because of a software patent. In this presentation, we more rigorously apply 35 USC 112(f) in accordance with the proposal Mark Lemley outlines in his WIRED oped "Let's Go Back to Claiming the Problem Not the Solution" to 30 patents - 10 PAE and 20 control patents, provided by Patent Freedom. We find that 1) PAE patents are overwhelmingly functionally claimed (100%), but non-PAE patents are also functionally claimed (50%), 2) a very high share of the PAE patents contained claims whose elements were supported only by the highest levels of abstraction, and 3) that not all code is created equal – "detailed" code over generic elements does not necessarily promote technical progress. Our findings suggest that significant numbers of high impact patents could have their broadest claims knocked out for lack of support under 112(b) if functional claiming, short of the magic "means for" language were recognized more broadly and scrutinized more meaningfully. But the courts and the PTO would need to do so.

Executive Summary

There is a perception that impossibly broad software patents are responsible for much of what ails the patent system. Patents that claim making an electronic version of a document,[4] or connecting wirelessly to the internet have provoked anxiety[5] in small business owners and public concern that the patent system is harming rather than promoting innovation. We find some justification for worrying about software patents – applying the Graham-Vishnubhakat

[1] Based on testimony given at the February 12, 2013, USPTO Silicon Valley Software Partnership Forum at Stanford Law School. Copy of presentation provided at Appendix D.

[2] Assistant Professor, Santa Clara University School of Law, B.S. Engineering, Stanford, Reg #55,062. This comment draws from Professor Chien's experience prosecuting patents, including software and hardware patents, prior to becoming a professor. The authors thank Patent Freedom, RPX Corporation, and Gazelletech for providing patent data and analytical support services.

[3] Research Assistant to Professor Colleen Chien and 2013 JD graduate, Santa Clara University School of Law, B.S. Computer Engineering, Virginia Tech, Reg #65,572. This comment draws from Aashish's experiences examining (AU3714) and prosecuting patents, including software and hardware patents.

[4] U.S. Patent No. 6,185,590 ("Process and architecture for use on stand-alone machine and in distributed computer architecture for client server and/or intranet and/or internet operating environments")

[5] Ars Technica, "Patent trolls want $1,000—for using scanners" (http://arstechnica.com/tech-policy/2013/01/patent-trolls-want-1000-for-using-scanners/)

definition of software patent[6] to patents in the RPX Litigation Database[7] we find that as many as 82% of all non-practicing entity or "troll"[8] defendants have been sued on the basis of a software patent, as compared to only 30% of non-NPE defendants to patent litigation suits.

To the problem of overly broad patents, however, Mark Lemley has a solution: take the most problematic patents, patents that through functional claiming seem to cover the broader "problem" rather than the narrower "solution," and limit their scope, to the actual disclosed implementation or its equivalents.[9] If his fix works, many claims should fail, and others will be narrowed.

But does it work? We tested Lemley's suggestion by looking for evidence of functional claiming[10] in 10 famous or high-impact PAE patents.[11] Excluding explicit means-plus-function claims, we found that some 70% of these high-impact patents contained claims that would likely fail because they contained claim elements that were support only at the highest level of abstraction. Others claims had greater support for their elements, e.g., native code or source code, and that half of the control patents, because they were also functionally claimed, would be subject to a more rigorous review.

In conclusion, our analysis validates the promise of more rigorously applying 35 USC 112(f) – all studied patents in the high-impact group contained functional claims. When we looked more closely at the specifications, we found that many lacked supported for one or more of the elements of the functional claims. While this effort is only exploratory, it validates the promise of more rigorously applying the law – many existing claims will fall away as unsupported.

[6] Stuart J. H. Graham & Saurabh Vishnubhakat, *Of Smart Phone Wars and Software Patents*, 27 Journ. of Ec. Persp: 1 (Winter 2013, 67–86) at fn 7. and p.75. (To make this determination, "Patent Office experts examined all US patent classes and subclasses and determined which were likely to contain patents applications or issued patents containing some element of either general purpose software or software that is specific to some form of hardware," resulting in a definition that the authors describe as "over-inclusive" and "under-inclusive.")

[7] A proprietary database of PAE and other NPE litigations maintained by RPX corporation and used and described in previous analysis.

[8] A term we apply colloquially to patent assertion entities, whose "use patents primarily to obtain license fees rather than to support the development or transfer of technology" Colleen V. Chien, From Arms Race to Marketplace: The Complex Patent Ecosystem and Its Implications for the Patent System, 62 HASTINGS L.J. 297, 326 (2010).

[9] Mark A. Lemley, *Functional Claiming and Software Patents*,____ Wisc. Law. Rev. 13 (2013)(Forthcoming)

[10] How we identified functional claims is described below, we exclude explicit means plus function claims from the count throughout this comment.

[11] To do so we developed a five-level framework for analyzing disclosure : functional abstraction (what a software program will do), abstract data type (a collection of data and set of operations on them), pseudocode (a set of instructions that specifies the operations that collectively achieve a function), data structure (a programming language construct that stores a collection of data), and source code.

If courts move to construe functional claims more rigorously, however, several realities may inform their approach. First, application of the rule in every context may be over inclusive because the boundary between functional and nonfunctional language is heavily dependent on the technology involved. Second, greater clarity, through court decisions, would be needed to define the scope of "supported" functional claims – i.e. the "equivalents" of psuedocode or source code. Third, examiners and applicants would need time and possibly support before the effective date of a PTO regimen that applies greater scrutiny to functional claims, especially in view of the importance of the original application disclosure in determining whether functional language should be rejected under our framework.

The following sections expand on this summary.

The Proposition

Mark Lemley's proposal[12] for fixing "most of the software patent problem" is to reign in overbroad patents by subjecting more of them to the limitations of 35 USC Section 112(f). "Functionally claimed" patents, he believes, should be curtailed in two ways: (1) if they don't contain adequate structure, they should be invalidated as unsupported under 35 USC Section 112(b); and (2) when structure is disclosed, the claim should be read to cover only the disclosed structure and its equivalents. Overbroad claims should thus fall away or cease to matter as easy to circumvent.

What We Did To Test the Proposition

To test the proposition, we: (1) identified relevant patents, (2) looked at their functional claims, and (3) analyzed their specifications for support for the functional claims using a technical abstraction framework we developed based on real-world programming constructs. We also undertook additional analyses to determine whether these patents would fare differently at the PTO if 35 USC 112(f) was applied against their functional claims.

<u>The Patents We Analyzed ("The Patent Freedom High PAE Impact and Control Patent Set")</u>

We analyzed thirty patents – ten high-impact patents and twenty control patents. With the help of Patent Freedom, a provider of information and analysis of NPE/PAE litigations, we picked 10 patents from high-profile campaigns (see Appendix A) roughly half that had been asserted against a large number of defendants in litigation and the remainder which we hand-picked; as well as a matched set of 20 non-NPE litigated patents, roughly half of which were among the most asserted, and the other half which were picked by Patent Freedom at random. While all thirty patents were analyzed for the presence of functional claiming, we carried out

[12] Mark A. Lemley, *Functional Claiming and Software Patents*,____ Wisc. Law. Rev. 3-4 (2013)(Forthcoming)

the second part of the analysis – applying the technical abstraction framework – on the high impact patents only.

Functionally claimed elements contain three elements: a computing element, functional triggering language, and a function associated with the computing element.[13] We looked for such claims and disregarded from our analysis claim elements reciting features in traditional "means-plus-function" form to focus on whether functional claiming analysis would change patentability outcomes independently of traditional analysis under 112(f).

The Computing Element

A computing element can be either a generic piece of computing hardware or a generic software construct. To minimize the risk of being underinclusive, we identified all computing elements associated with functional language; both generic hardware and hardware drawn to specific tasks were identified. An incomplete and open list of examples of computing hardware we identified include a 'processor,' 'memory,' 'user interface,' 'client,' 'controller,' and 'accelerometer.' A computing element need not be totally unbounded as a matter of structure, unlike traditional "means-plus-function" analysis under 112(6).[14] Similarly, generic software constructs may include a 'module,' 'unit,' 'engine,' 'interface,' and 'manager.' Determining whether the computing element is generic ultimately depends on whether that computing element would be considered generic to one of ordinary skill in the art at the time of invention.

The Triggering Language

Functional triggering language ties a computing element to its associated action. We found two categories of functional triggering language in the claims we analyzed. Triggering language describing a general capability was the more common of the two types, and includes commonly examined phrases under traditional "means-plus-function" claiming such as 'adapted to,'[15] 'for', 'capable of,' 'configured to,' 'programmable means for,' '…capable of engaging,' 'operable to' and 'for …ing.'[16] Structurally unbounded terms, including 'mechanism for,' module for,' 'device for,' unit for,' 'component for,' 'element for,' 'member for,' 'apparatus for,' 'machine for' or 'system for,' may still trigger 112(6) where means-for language is missing.[17] However, triggering language need not be limited to the formalistic phrases above. For example, a simple 'that' could be triggering language, as in a claim reciting "a runtime

[13] Mark A. Lemley, *Functional Claiming and Software Patents*,____ Wisc. Law. Rev. 20-21 (2013)(Forthcoming)
[14] Manual of Patent Examination and Procedure ("MPEP") 2181(I)(A)
[15] MPEP 2173.05(g)
[16] Mark A. Lemley, *Functional Claiming and Software Patents*,____ Wisc. Law. Rev. 18 (2013)(Forthcoming)
[17] MPEP 2181(I)(A)

engine that invokes said at least one interface object to access data from the relational database."[18]

The Function

An action associated with the computing element is a 'function' if the action describes a goal, rather than a concrete part of achieving that goal. For example, "identifying the color of a block" may be considered functional for defining a goal without defining how that goal is reached, whereas "identifying a detected wavelength of light reflected from a block" may not be functional for defining a discrete action that one of ordinary skill in the art would recognize as a concrete part of "identifying the color of a block."

Method claims that recite functional features are analogous to "step-plus-function" claims in the same way that functionally claimed elements reciting a computing element are analogous to "means-plus-function" claims. However, method claims that recite functional language commonly recite no computing element and no functional triggering language. Using the block example above, a method may recite only a step for "identifying the color of a block" as one of its elements. For such method claims reciting functional language only, we analyzed the element for functional characteristics and disregarded the lack of computing element and functional triggering language.

Technical Abstraction Framework

Software may be described at many levels, each striking varying balances between software principles applied to solving a problem and the actual collection of computer instructions applied to solve that problem. By describing software at multiple levels, software engineers define abstract problems, divide those abstractions into manageable portions, and develop software code implementing those portions to solve the abstract problem at hand. By imagining software at various levels of abstraction, software engineers easily move between a 30,000 foot view of the project as a whole and a ground level view of code to create elegant and efficient software solutions to complex problems.

[18] U.S. Patent No. 6,101,502 at claim 10.

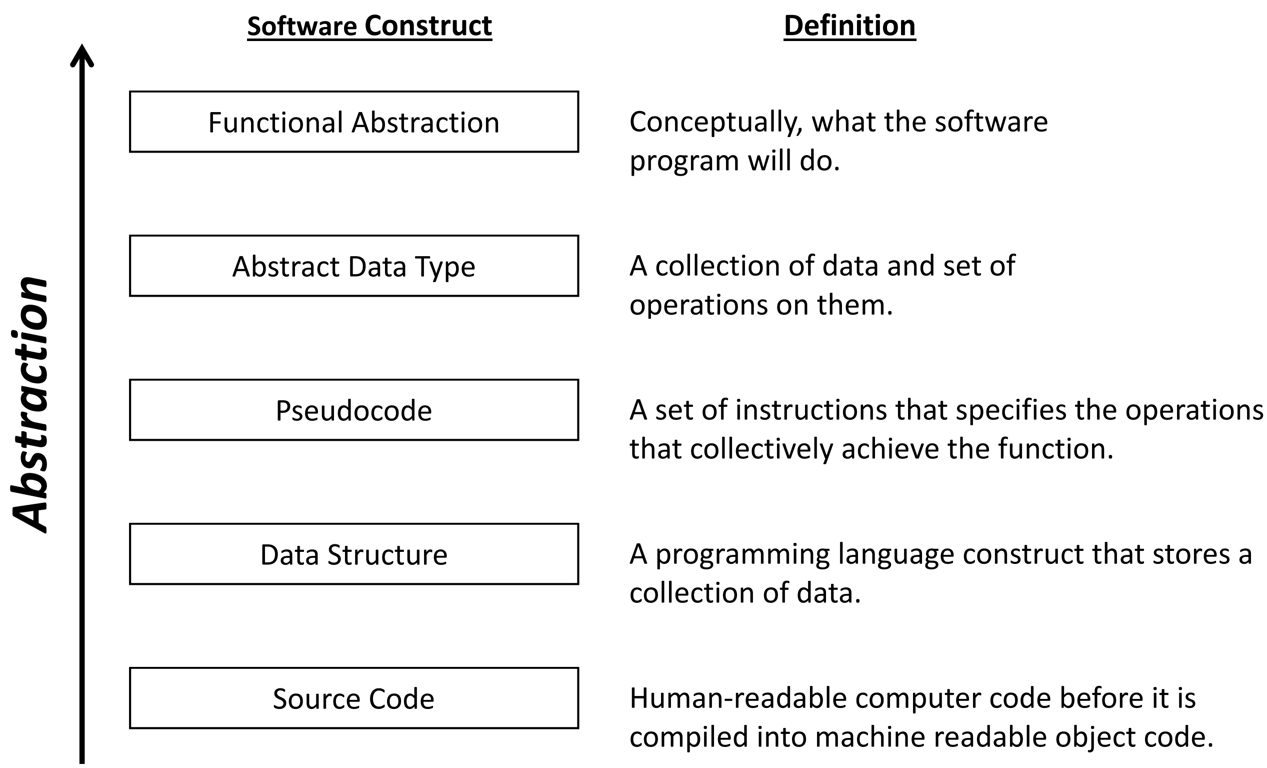


Fig. 1: Overview of Abstraction Levels

Real-world software abstraction techniques are also applicable in the patent context, to identify whether functional language is sufficiently supported in a particular patent disclosure. These real-world software engineering principles form the foundation of our framework for identifying functional claiming.[19] We categorized these abstraction levels into five levels. At the highest level, functional abstraction defines a problem and goal and is analogous to functional language in the patent context. At the two levels of abstraction directly below functional abstraction, Abstract Data Types and Pseudocode describe a solution independently of any computing infrastructure. Finally, Data Structures and Source Code form the two lowest levels of abstraction, and describe a solution tied to a specific computing infrastructure. With each step down, the patentee discloses more detail about the invention, narrowing the scope of the disclosure.

Stating the Goal: Functional Abstraction

Functional Abstraction is the highest level of abstraction, defining in conceptual terms a desired end result for solving a particular problem. At this level, only the end goal to be reached is disclosed; functional abstraction only expresses the goal to be reached, without how

[19] Carrano, Frank and Prichard, Janet, Data Abstraction and Problem Solving with C++, 3rd. Ed.

that goal is to be reached. For example, a Functional Abstraction for transportation would simply be "getting from point A to point B." Lemley noted[20] that this level of abstraction, describing the "what" without the "how," as among the most troubling problems with software patent claims.

Conceptualizing the Solution: Abstract Data Types

The two abstraction levels below Functional Abstraction conceptually specify how a solution is reached, without reference to specific computing infrastructure. These abstraction levels, including Abstract Data Types and Pseudocode, allow the structure of software to be described in detail without limiting a solution to a particular computing infrastructure.

Abstract Data Types describe software at the second-highest level of abstraction, addressing in broad strokes "how" the goal is to be reached. In modern software engineering, software is commonly organized into one or more collections of data and an attendant set of operations on that data to form an ecosystem that, in unison, accomplishes a goal. The task of defining Abstract Data Types that can accomplish the goal of a particular Functional Abstraction commonly requires the most challenging conceptual leap in software engineering. From the transportation example above, Abstract Data Types may include as its primary mode of transport wings, wheels, propellers or paddles. Because a clever engineer likely would not include all of these possibilities in one transport, Abstract Data Types also help to define more efficient solutions.

Defining the Nuts and Bolts: Pseudocode

Pseudocode is, at the third-highest level of abstraction, a detailed description of operations that collectively achieve a specific solution. Pseudocode describes a solution in great enough detail that it may be implemented as source code in a desired computer language with modifications for compliance with that language's syntax or quirks. In a common software engineering methodology, individual sets of operations defined in an Abstract Data Type may be fleshed out pseudocode to allow direct mapping of particular steps into a specific programming language. It is important to note, however, that even though pseudocode may be transformed into a specific programming language in a straightforward fashion, pseudocode itself is written largely without respect to any specific programming language. Continuing the transportation example, an Abstract Data Type for a wing may further include Pseudocode describing how cables inside the wing actuate individual lifting surfaces. Pseudocode may, then, define the "guts" of an Abstract Data Type, allowing its full functionality to become apparent.

---

[20] Mark A. Lemley, *Functional Claiming and Software Patents*,____ Wisc. Law. Rev. 18 (2013)(Forthcoming)

Building the Skeleton: Data Structures

The lowest two abstraction levels disclose actual implementation of a solution on a particular computing infrastructure. These abstraction levels are Data Structures and Source Code, each implemented in a specific computer language or other construct that is implementable in a computing infrastructure as-is. Of course, supporting computing infrastructure known to those skilled in the art, like particular operating system or supporting library code, may not always need to be disclosed.

Data Structures are programming language constructs implementing Abstract Data Types. The Data Structure may be considered a combination of Abstract Data Types and Pseudocode as translated into a particular computer language. The Data Structure combines the objects defined as Abstract Data Types and their internal operations as described into Pseudocode into a single construct able to perform all of the functions of the Abstract Data Type in a specific computing infrastructure.

Fleshing It Out: Source Code

Source Code is human-readable computer code commonly in the form of a computer language. Source Code is the text-based implementation of the solution in machine code as developed by humans. A multitude of languages in which Source Code may be developed exist, ranging from the classical (FORTRAN, Lisp, C) to the modern (Python, Ruby, Javascript). Source Code is transformed into machine-readable Object Code, or Machine Code, for execution on a specific computing infrastructure. Some patents may include Object Code[21] in addition or in place of Source Code. The meaningfulness of Object Code to one of ordinary skill in the art depends on whether the patent includes sufficient disclosure of the processor on which the object code runs; the object code cannot be implemented without knowledge of the processor for which the object code is tailored.

Abstraction levels can provide signposts pointing to both to the depth and the breadth of disclosure of software-related patentable solutions. With multiple abstraction levels, patentees would have greater information regarding the sufficiency of a particular disclosure, and examiners would have an organized framework for determining the scope of functionally claimed features. Unlike a binary determination of functional vs. nonfunctional support, abstraction levels allow discerning support for software claims at a higher level of granularity than otherwise possible.

Our functional claiming framework is compatible with, but more thorough than, existing patent examination procedure for determining scope of computer-implemented functional

[21] U.S. Patent No. 6,150,947 at col. 7, lines 17-25; cols. 9-10.

language. Specifically, while existing patent examination procedure allows a claim to be rejected where equivalent computing hardware is available as prior art,[22] we propose a more detailed analysis of the function itself to determine whether the patent's disclosed implementation defines meaningful limitations on that function.

Additional Analyses

Finally, to answer the question, would these patents fare differently at the PTO if 35 USC 112(f) was applied their functional claims, we considered each patent as a whole. We looked in particular for the presence of traditional means plus function claims and the presence of non-functional claims within the patent. If functional claims were present, we looked at the prosecution history for evidence of 112(f)/112(b) examination. They should be treated the same as means plus function claims in accordance with best prosecution practice.[23]

What We Found

Functional claiming was universal in our high-impact patents, but we found only bare-bones disclosure for functional elements in many of those patents. Though not universal, we found functional claiming to be common in our control patents. Our functional claiming framework did eliminate some claims, while allowing others to survive with narrower scope based on the level of abstraction disclosed in the patent.

Functional Claiming is Prevalent Among High Impact Patents, but not non-High Impact Patents

We found functional claiming to be prevalent in all of the high impact patents we studied. Notably, 100% of the analyzed high impact patents included at least one functional claim element. In contrast, we found that 40% of the control group patents are directed to software and recite functional claim elements. When non-software patents were excluded from the control group, the share of patents reciting functional claim elements also rose to 100%. Though functional claiming is less common in the control group patents we analyzed, software-related patents in both groups disproportionately recite functional claim elements.

We found no "means-plus-function" language in any software patents we analyzed, in either the high impact group or the control group. Such language did appear, however, in 15% of the control group patents, all of which are directed to mechanical or electromechanical

[22] See MPEP 2114(IV).
[23] MPEP 2181(I.A)

systems. The software patents we analyzed thus avoid the traditional means-plus function analysis that depends largely on linguistic formalisms.[24]

Computing Element Examples

The analyzed patents recite a wide range of computing elements including a processor,[25] a user interface,[26] a mapping routine,[27] a digital signal processor,[28] a search engine,[29] a satellite receiver,[30] and a management system,[31] as computing elements, among others. The computing elements we identified range from the tangible to the fanciful, but we selected them all because they signify an entity for performing a subsequently claimed action.

Functional Claiming Examples

Among the high-impact software patents analyzed, we identified many broad claim elements through our functional claiming framework. Particularly interesting are claim elements that describe expansive goals but provide no indication of the underlying implementation making that goal possible. Some notable functional claim elements lacking support below the functional abstraction level include "a user interface … configured to elicit, from a user, information about the user's perception of the commodity,"[32] a "network distribution rule to manage one or more system resources,"[33] and "a routing processor configured to determine if the media switch can stream media for the request."[34] With this language identified, we could target our search in the disclosure to finding support for a subset of claimed features, rather than the entire claim.

Functional Claims Commonly, But Not Always, Contained One or More Unsupported Elements

Not all patent disclosures are created equal, and the high-impact patents we studied varied greatly in the amount of support for functional claims. Though many patents have support only at the functional abstraction level, others include support for functionally claimed features at multiple levels of abstraction. In more complex instances, only a subset of features of functional claims found support below the functional abstraction level. Of the high impact patents we studied, only 30% had claim elements that were fully supported below the

[24] MPEP 2181(I)(A).
[25] U.S. Patent No. 7,222,078 at claim 1.
[26] U.S. Patent No. 7,222,078 at claim 1.
[27] U.S. Patent No. 7,346,472 at claim 11.
[28] U.S. Patent No. 6,150,947 at claim 19.
[29] U.S. Patent No. 5,930,474 at claim 1.
[30] U.S. Patent No. 5,223,844 at claim 12.
[31] U.S. Patent No. 8,015,307 at claim 1.
[32] U.S. Patent No. 7,222,078 at claim 1.
[33] U.S. Patent No. 8,015,307 at claim 1.
[34] U.S. Patent No. 7,054,949 at claim 1.

functional abstraction level by the corresponding patent disclosure. A significant fraction of patent claims surviving functional claiming analysis in some form, though the scope of a particular claim element may be significantly narrower when read in view of its supporting disclosure.

Determining support for software claims requires a nuanced review of every claimed feature and of the entire patent disclosure. Simply identifying support below functional abstraction anywhere in the patent disclosure is not enough; in many cases, high impact patents included support for known features at lower levels of abstraction but failed to disclose functional support for key inventive features.

Not all software patents are so easily dismissed; many disclose a great deal of support at many abstraction levels. Notably, U.S. Patent No. 5,930,474, asserted by Geotag (the "Geotag patent") includes robust disclosure at every level of functional abstraction. Directed to delivering info "such as business services, entertainment, news, consumer goods" for a user's local area, the Geotag patent starts with a functional abstraction[35] and describes inventive features at all lower levels. Abstract Data Types are illustrated as databases organizing information,[36] and Pseudocode for search subroutines are disclosed in narrative form.[37] Implementations are also disclosed, with Data Structures storing HTML information[38] and Source Code for HTML web pages.[39] The Geotag patent provides instructive examples of support for functional claims at every level of abstraction.

[35] U.S. Patent No. 5,930,474 at col. 7, lines 5-29.
[36] U.S. Patent No. 5,930,474 at fig. 2C.
[37] U.S. Patent No. 5,930,474 at col. 12, lines 35-45.
[38] U.S. Patent No. 5,930,474 at fig. 20.
[39] U.S. Patent No. 5,930,474 at cols. 27-28.

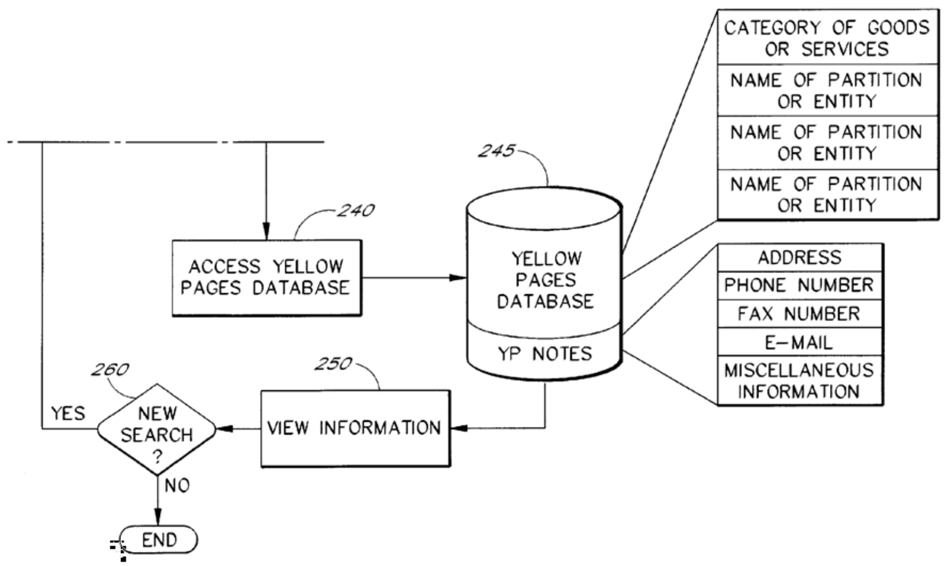


Fig. 2: Abstract Data Types in the Geotag Patent (Fig. 2C)

In the most complex situations, a patent disclosure may appear generally to provide support for functional claims but lack support for each and every functionally claimed element. A thorough analysis of each claim element is required to ensure that the patent disclosure supports all functionally claimed elements below the level of functional abstraction. Though U.S. Patent No. 6,185,590, asserted by Project Paperless (the "Project Paperless patent"), discloses Source Code for certain error-detection features (*),[40] while leaving more complex mapping functions disclosed only as functional abstractions (bold**):[41]

Loading and unloading the engine (DLLs provided into and out of memory)*

Mapping original functions to engine object counterparts**

Adding general error detection and correction*

Determining and matching arguments and return values** for mapping the original functions to their engine object counterparts In order to add assertion and error detection and correction, the original function must be wrapped and called from within the engine object version of the original function.

Managing error feedback. All APIs have their own way providing error feedback. Since one of the goals of the Engine Management layer is to generically manage

[40] U.S. Patent No. 6,185,590 at cols. 15-16.
[41] U.S. Patent No. 6,185,590 at col. 17, lines 29-50.

error detection, correction, and feedback, it must handle all errors identically … **By creating specific classes of APIs the process of generating Layer 1 engine management may be expedited manually and/or automatically****."

U.S. Patent No. 6,185,590 at col. 17, lines 29-50.

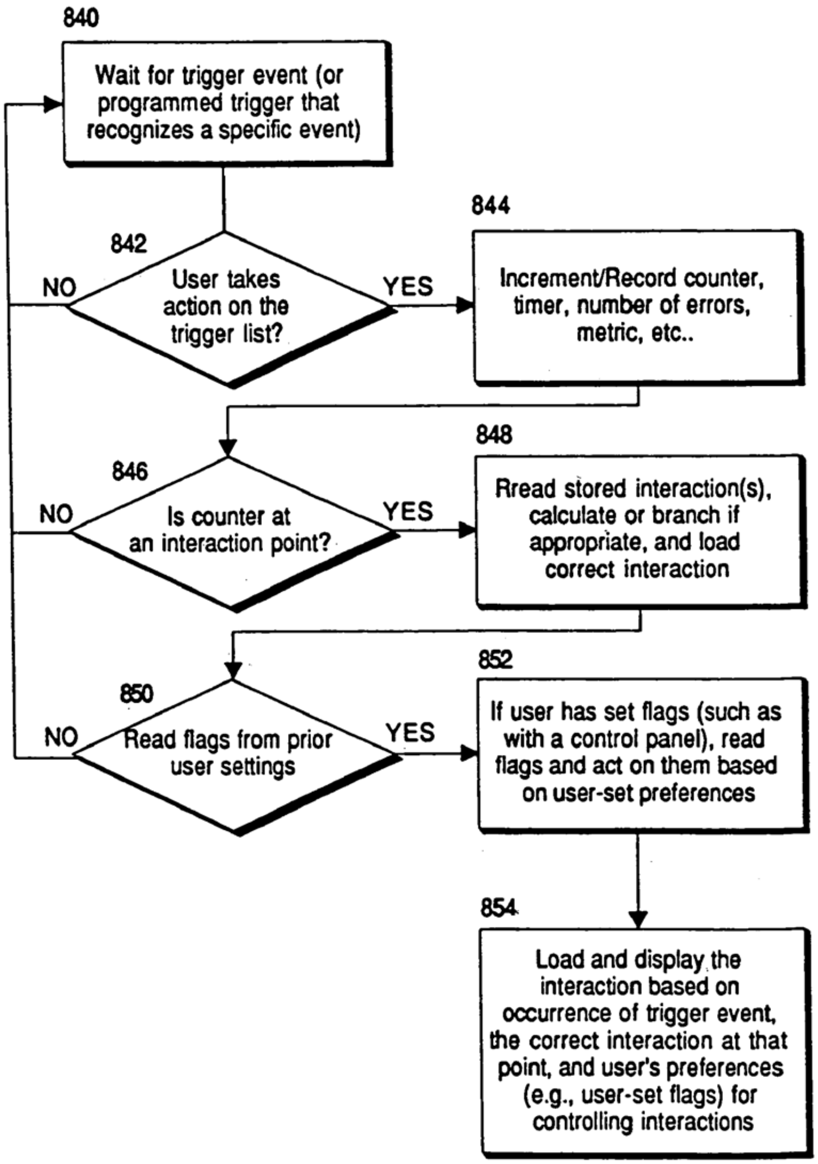
Similarly, although U.S. Patent No. 7,222,078, asserted by Lodsys (the "Lodsys patent"), does include some discussion of Pseudocode,[42] many functionally claimed features lack support below functional abstraction. For example, the Lodsys patent includes a flowchart describing a process for handling a user interaction. At steps 844, 848, 852, and 854, the process executes various actions. Only step 844, however, describes any action at a level of detail greater than function abstraction. Steps 848, 852, and 854, though mixed together with Pseudocode, are themselves functional abstractions.

Conclusion

To conclude, our analysis validates the promise of more rigorously applying 35 USC 112(f) – all studied patents in the high-impact group contained functional claims. When we looked more closely at the specifications, we found that many lacked supported for one or more of the elements of the functional claims. While this experiment is only exploratory, its result shows the promise of more rigorously applying the law – many existing claims will fall away as unsupported.

If courts move to construe functional claims more rigorously, as we believe they should, several realities should be kept in mind. First, application of the rule in every context may be over inclusive because the boundary between functional and nonfunctional language is heavily dependent on the technology involved. Second, greater clarity, through court decisions, would be needed to define the scope of "supported" functional claims – i.e. the "equivalents" of psuedocode or source code. Third, examiners and applicants would need time and possibly support before the effective date of a PTO regimen that applies greater scrutiny to functional claims, especially in view of the importance of the original application disclosure in determining whether functional language should be rejected under our framework.

[42] U.S. Patent No. 7,222,078 at fig. 23, elems. 844, 846, 850, 852.



Pseudocode for Handling of Variables

Functional Abstraction of Decision-making for User Interaction

Functional Abstraction of Decision-making using User Preferences

Functional Abstraction of Action controlled by User Preferences

<u>Fig. 3: Mixing Functional Abstraction and Pseudocode in the Lodsys Patent (Fig. 23)</u>

We found that, even among the high impact patents we studied, the level of support for functional claims varied widely from patent to patent. Both U.S. Patent Nos. 7,054,949 and 8,105,307, asserted by Single Touch, only disclose functional abstractions, and occupy the opposite end of the spectrum from the relatively well-supported Geotag patent.[43] Both the Lodsys and Project Paperless patents highlight the criticality of identifying support for each claim element in its accompanying disclosure to determine whether a particular claim as a whole is supported beyond only functional abstraction.

[43] U.S. Patent No. 5,223,844 at col. 27, lines 27-32.

Examiners Rarely Reject Claims Reciting Functional Language

We found that, among the high-impact patents we studied, examiners rarely rejected claims under 112(f). Of the five high impact patents with publicly available file histories, we found no rejections of functional language, and only one acknowledgement of the existence of functional language,[44] in the prosecution history of the Lodsys patent. The examiner noted certain claimed subject matter as functional under the doctrine of "intended use"[45] rather than within the scope of 112(f), because the claims recite no formalistic "means for" triggering language, as is required for any rejection under 112(f).

[44] U.S. Patent Application Serial No. 10/734,102, Non-Final Rejection dated March 29, 2005 at pp. 2-3.

[45] MPEP 2114(I),(II),(IV) (stating that an apparatus claim element is not patentable if the equivalent apparatus, irrespective of whether its function (or "intended use" as recited by claims under examination), exists in prior art).

Appendix A: The Patents We Analyzed ("The Patent Freedom High Impact PAE and Control Patent Sets")

High-Impact PAE Patents

Patent No.	Campaign	Defendants by Campaign	FL*	DS*	DS*
7,222,078	Lodsys	106	Y	N	Y
7,346,472	Bluespike	79	Y	Y	N
5,937,402	Datatern	70	Y	Y	Y
6,101,502			Y	Y	N
5,930,474	Geotag	435	Y	N	Y
6,150,947	Ogma	32	Y	Y	N
5,223,844	PJC Logistics	281	Y	N	Y
6,185,590	Project Paperless	3	Y	N	Y
7,054,949	Single Touch	1	Y	N	Y
8,015,307			Y	N	Y

*FL: whether functional language appears in at least one claim

*DS: whether any claim element was unsupported (e.g. supported only at the functional abstraction level)

Control Patent Group

Patent No.	Campaign	Technology	FL*
Re. 40,081	FastVDO LLC	Electronics	Y
6,128,454	Canon Inc	Mechanical	Y
5,742,737	eDigital Corp	Software	Y
7,933,122	Otter Products LLC	Mechanical	N
8,156,944	Ruyan Investment Holdings Ltd	Mechanical	N
8,135,122	NobelBiz Inc	Software	Y
5,560,360	Neurografix	Biotech / Pharmaceutical	Y
6,462,713	Transdata Inc	Mechanical	N
7,627,975	Prototype Productions Inc	Mechanical	N
7,742,084	Eastman Kodak Co	Software	Y
8,088,480	Shieldmark Inc	Mechanical	N
6,763,998	United Coin Machine Co	Mechanical	Y
7,931,199	Serverside Group Limited	Software	Y
7,982,720	Immersion Corp	Software	Y
6,293,556	Krausz Industries	Mechanical	N
6,722,686	Cequent Performance Products Inc	Mechanical	Y
6,400,376	Ericsson Inc	Software	Y
8,071,577	Bayer Pharma AG	Biotech / Pharmaceutical	N
5,949,880	Maxim Integrated Products Inc	Software	Y
7,139,974	Bascom Research LLC	Software	Y

*FL: whether functional language (including "means-plus-function") appears in at least one claim

Appendix B: Our Framework for Analyzing the Support Within Patents

See Figure 1.

Appendix C: Analysis of the High-Impact Patents

U.S. Patent No. 7,222,078 owned by Lodsys (Claim 1)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A system comprising: units of a commodity that can be used by respective users in different locations, **a user interface**, which is part of each of the units of the commodity, **configured to provide a medium for two-way local interaction between one of the users and the corresponding unit of the commodity,**	**functional abstraction** at col. 10, lines 1-14
and further **configured to elicit, from a user, information about the user's perception of the commodity,**	**functional abstraction** at col. 31, lines 48-64
a memory within each of the units of the commodity **capable of storing results of the two-way local interaction,**	**functional abstraction** at col. 22, lines 56-67
the results including elicited information about user perception of the commodity, **a communication element** associated with each of the units of the commodity **capable of carrying results of the two-way local interaction from each of the units of the commodity to a central location,** and	**functional abstraction** at col. 41, lines 21-54 and col. 58, lines 6-60
a component capable of managing the interactions of the users in different locations and	**functional abstraction** at col. 5, lines 6-60
collecting the results of the interactions at the central location.	**functional abstraction** at col. 60, lines 14-27

U.S. Patent No. 7,346,472 owned by Bluespike (Claim 11)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A computerized system for monitoring and analyzing at least one signal: **a processor that creates an abstract of a signal using selectable criteria**;	**abstract data type** at col. 9, lines 55-61
a first input that receives at least one reference signal to be monitored, said first input being coupled to said processor such that **said processor may generate an abstract for each reference signal input to said processor**;	**pseudocode** at col. 10, lines 9-33
a reference database, coupled to said processor, that stores abstracts of each at least one reference signal; a second input that receives at least one query signal to be analyzed, said second input being coupled to said processor such that **said processor may generate an abstract for each query signal**;	**abstract data type** at col. 11, lines 24-31
a **comparing device**, coupled to said reference database and to said second input, that **compares an abstract of said at least one query signal to the abstracts stored in the reference database to determine if the abstract of said at least one query signal matches any of the stored abstracts,**	**abstract data type** at col. 8, lines 55-67 and col. 9, lines 1-10
wherein **the comparing device identifies at least two abstracts in the reference database that match the abstract of said at least one query signal and an index of relatedness to said at least one query signal for each of said at least two matching abstracts.**	**abstract data type** at col. 11, lines 13-23

U.S. Patent No. 5,937,402 owned by Datatern (Claim 17)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A system for enabling access to a relational database from an object oriented program, comprising: **a normalization process for inputting one or more denormalized relational schema objects,**	**data structure** at col. 4, lines 24-52
said set of one or more denormalized relational schema objects corresponding to a physical table segmented into rows and columns, said normalization process further forming a normalized schema object, responsive to said set one or more denormalized relational schema objects, said normalized relational schema object representing a logical table comprising a subset of said columns of said physical table; and **a mapping process for generating, responsive to said normalized relational schema object, one or more object classes associated with said normalized relational schema object.**	**functional abstraction** at col. 10, lines 66-67 and col. 11, lines 1-19
U.S. Patent No. 6,101,502 owned by Datatern (Claim 10)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A computer program fixed on a computer-readable medium and adapted to operate on a computer to provide access to a relational database for an object oriented software application, comprising: **a mapping routine that generates a map of at least some relationships between schema in the database and a selected object model;**	**abstract data type** at Table 1 and Table 3
a code generator that employs said map to create at least one interface object associated with an object corresponding to a class associated with the object oriented software application; and	**data structure** at col. 6, lines 31-64
a runtime engine that invokes said at least one interface object to access data from the relational database.	**data structure** at col. 6, lines 8-30

U.S. Patent No. 5,930,474 owned by Geotag (Claim 1)	
Claim Element	**Supporting Disclosure for Functional Lang.**
1. A system which associates on-line information with geographic areas, said system comprising: a computer network wherein a plurality of computers have access to said computer network; and an organizer executing in said computer network, wherein said **organizer is configured to receive search requests from any one of said plurality of computers,**	**functional abstraction** at col. 6, lines 46-67 and col. 7, lines 1-4
said organizer comprising: a database of information organized into a hierarchy of geographical areas wherein entries corresponding to each one of said hierarchy of geographical areas is further organized into topics; and a search engine in communication with said database, said **search engine configured to search geographically and topically,**	**abstract data type** at col. 19, lines 29-63; col. 22, lines 39-67; and col. 23, lines 1-3
said **search engine further configured to elect one of said hierarchy of geographical areas prior to selection of a topic so as to provide a geographical search area wherein within said hierarchy of geographical areas at least one of said entries associated with a border geographical area is dynamically replicated into at least one narrower geographical area,**	**abstract data type** at col. 19, lines 29-63
said **search engine further configure to search said topics within said selected geographical search area.**	**pseudocode** at col. 22, lines 39-67 and col. 23, lines 1-3

U.S. Patent No. 6,150,947 owned by Ogma (Claim 2)	
Claim Element	**Supporting Disclosure for Functional Lang.**
The programmable motion-sensitive sound effects device as claimed in claim 1 wherein said motion-sensitive actuator further comprises **a sound effect storage for storing at least one predetermined sound effect** and	**pseudocode** at col. 4, lines 2-6
wherein the **function of the acceleration used to calculate the numerical values is a derivative of the acceleration in each of the coordinate axes.**	**pseudocode** at col. 6, lines, 23-28
U.S. Patent No. 5,223,844 owned by PJC Logistics (Claim 12)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A mobile unit for a vehicle monitoring system, comprising: **a vehicle condition sensor for generating signals varying with the operation of the vehicle;**	**pseudocode** at col. 14, lines 32-68 and col. 15, lines 1-46
an operator activated sensor for generating signals identifying an operator input message;	**pseudocode** at col. 20, lines 51-56
a satellite receiver responsive to satellite position information including latitude, longitude and time, the satellite receiver generating vehicle position signals correlated to a received time; a cellular telephone transmitter for transmitting information onto a cellular telephone communications link; and **a mobile unit controller responsive to signals varying with the operation of vehicle [sic],**	**functional abstraction** at col. 22, lines 38-50
signals [sic] identifying an operator input message and the vehicle position signals, **the mobile unit controller transmitting** signals from the cellular telephone transmitter **in accordance with a priority designation between the signals varying with operator inputs along and simultaneous therewith the vehicle position signals.**	**functional abstraction** at col. 22, lines 51-68 and col. 23, lines 1-15

U.S. Patent No. 6,185,590 owned by Project Paperless (Claim 1)	
Claim Element	**Supporting Disclosure for Functional Lang.**
A distributed computer implemented process for migrating at least one program specific Application Programmer Interface (API) from an original state into a substantially consistent interface by building an object for at least one of an engine and a viewer process, the object providing substantially uniform access to the at least one of the engine having engine settings and the viewer process, comprising the steps of: (a) providing, on a server, the at least one engine and viewer process, each with one or more features to be executed; (b) providing, on at least one of the server and another server connectable to the server, **at least one engine component or another viewer process configured to execute the one or more features by converting the at least one program specific Application Programmer Interface (API) from the original state into the substantially consistent interface,**	**functional abstraction** at col. 14, lines 32-65
and mapping the substantially consistent interface to the at least one of the engine and the viewer process; and	**functional abstraction** at col. 15, lines 57-67
(c) providing, on **a client configured to be connectable to the server and optionally configured to be connectable to the another server,**	**abstract data type** at col. 24, lines 25-42
an object manager layer communicable with and managing the at least one engine component or the another viewer process via the substantially consistent interface.	**functional abstraction** at col. 23, lines 20-31

U.S. Patent No. 7,054,949 owned by Single Touch (Claim 1)	
Claim Element	**Supporting Disclosure for Functional Lang.**
ystem for streaming media comprising: **a edia switch configured to** receive servation data for a request for media and receive a reservation identification, to **ocess the reservation identification and the servation data to determine if the servation identification is valid, and, if lid, to stream at least partially the media the request;**	**functional abstraction** at col. 8, lines 42-61
outing processor configured to receive the servation data, to **determine if the media itch can stream media for the request,** and transmit the reservation data to the media itch if the media switch is able, at least tially, to stream media for the request; and	**functional abstraction** at col. 11, lines 33-67 and col. 19, lines 43-67
management system configured to receive e request for media, to **build a reservation ving the reservation data and the servation identification for the request,** and transmit the reservation data to the routing ocessor.	**functional abstraction** at col. 23, lines 28-35
U.S. Patent No. 8,015,307 owned by Single Touch (Claim 1)	
Claim Element	**Supporting Disclosure for Functional Lang.**
ystem for streaming media to a viewer for a quest for media comprising: **media switch** to receive from the viewer at a edia switch address a reservation entification and a presentation entification, to receive reservation data mprising a valid reservation identification, **validate the reservation identification ing the valid reservation identification, d, if validated, to stream to the viewer at st some media for a presentation entified by the presentation identification,**	**functional abstraction** at col. 8, lines 42-61
e presentation comprising at least one edia identification and at least one network tribution rule, **the at least one network tribution rule to manage one or more stem resources;**	**functional abstraction** at col. 10, lines 21-33

U.S. Patent No. 8,015,307 owned by Single Touch (Claim 1) *continued*	
Claim Element	**Supporting Disclosure for Functional Lang.**
a routing processor comprising a routing processor identification and **configured to** receive from the viewer the presentation identification and the reservation identification at the routing processor identification, to receive the reservation data, to **use the presentation identification to identify the presentation, (C24L36-54) to select the media switch based on the at least one network distribution rule for the presentation,**	**functional abstraction** at col. 26, lines 19-40
to determine if the media switch is configured to stream the media for the presentation,	**functional abstraction** at col. 9, lines 14-35 and col. 27, lines 39-49
and, if so configured, to transmit the reservation data to the media switch and to transmit the media switch address to the viewer; and **a management system** to receive the request for media, **to build a reservation comprising the reservation identification to be validated at the media switch, the routing processor identification, and the presentation identification,**	**functional abstraction** at col. 9, lines 4-13
to reserve a resource to stream media for the reservation, to transmit the reservation to the viewer, and to transmit the reservation data to the routing processor.	**functional abstraction** at col. 22, lines 65-67 and col. 23, lines 1-5

Appendix D: Analysis of the Control patents

U.S. Patent No. Re. 40,081 owned by FastVDO LLC Claim 1: functional language present
An apparatus for coding, storing or transmitting, and decoding M.times.M sized blocks of digitally represented images, where M is an even number.Iadd., .Iaddend.comprising a. a forward transform comprising i. a base transform having M channels numbered 0 through M-1, half of said channel numbers being odd and half being even; ii. an equal normalization factor in each of the M channels selected to be dyadic-rational; iii. a full-scale butterfly implemented as a series of lifting steps with a first set of dyadic rational coefficients; iv. M/2 delay lines in the odd numbered channels; v. a full-scale butterfly implemented as a series of lifting steps with said first set of dyadic rational coefficients; and vi. a series of lifting steps in the odd numbered channels with a second specifically selected set of dyadic-rational coefficients; **b. means for transmission or storage of the transform output coefficients;** and c. an inverse transform comprising i. M channels numbered 0 through M-1, half of said channel numbers being odd and half being even; ii. a series of inverse lifting steps in the odd numbered channels with said second set of specifically selected dyadic-rational coefficients; iii. a full-scale butterfly implemented as a series of lifting steps with said first set of specifically selected dyadic-rational coefficients; iv. M/2 delay lines in the even numbered channels; v. a full-scale butterfly implemented as a series of lifting steps with said first set of specifically selected dyadic-rational coefficients; vi. an equal denormalization factor in each of the M channels specifically selected to be dyadic-rational; and vii. a base inverse transform having M channels numbered 0 through M-1.

U.S. Patent No. 6,128,454 owned by Canon Inc. Claim 1: functional language present
An electrophotographic image forming apparatus for forming an image on a recording material, comprising: an electrophotographic photosensitive drum; **charging means for charging said electrophotographic photosensitive drum;** **developing means for developing a latent image formed on said electrophotographic photosensitive drum into a toner image;** **transfer means for transferring the toner image formed by said developing means from said electrophotographic photosensitive drum onto said recording material;** **fixing means for fixing the toner image transferred onto the recording material by said transfer means on the recording material;** a motor; a driving rotatable member for transmitting a rotational driving force from said motor; wherein said driving rotatable member has formed therein a twisted hole at a central portion thereof having a non-circular cross-section with a plurality of corner portions; and a twisted projection provided at a longitudinal end of said electrophotographic photosensitive drum and having a non-circular cross-section with a plurality of corner portions, said twisted projection being engageable with the twisted hole, wherein the rotational driving force is transmitted to said electrophotographic photosensitive drum by engagement between the twisted hole and twisted projection, and wherein said twisted projection is urged toward said twisted hole when said driving rotatable member is rotated with said twisted projection being in engagement with the twisted hole.
U.S. Patent No. 5,742,737 owned by eDigital Corp. Claim 4: functional language present
A method for recording a new message on a hand held recording device without disturbing the physical continuity of existing messages and without manually searching for a blank segment of memory on the flash memory digital recording medium, said method comprising the steps of: **a) placing the recording device in an idle mode** where all recorder functions are inactive; and b) activating a record switch causing the recording device to: **i) search for an end of a last recorded message on the recording medium,** **ii) identify a segment of flash memory past the end of a last recorded message as a beginning point where the new message may be recorded, and** **iii) begin recording a new message at the beginning point.**

U.S. Patent No. 7,933,122 owned by Otter Products LLC Claim 1: NO functional language
A protective enclosure for a computer comprising: a flexible membrane that is molded to fit over at least a front portion of said computer that allows interactive access to controls on said front portion of said computer; a hard shell cover that fits over said flexible membrane and said computer and that is formed to provide openings that allow a user to access said flexible membrane to have interactive access to said controls of said computer, said hard shell cover providing rigidity to said protective enclosure, said hard shell cover comprising a front shell formed to a rigid shape of a front portion of said computer and a back shell formed to a rigid shape of a rear portion of said computer; a stretchable cushion layer that is disposed over said hard shell cover that has sufficient elasticity to substantially confirm to said hard shell cover and provide cushioning to said protective enclosure, said stretchable cushion layer exposing at least a portion of said hard shell cover and having a tab disposed to fit into a corresponding groove in said hard shell cover.
U.S. Patent No. 8,156,944 owned by Ruyan Investment Holdings Litd. Claim 1: NO functional language
An aerosol electronic cigarette, comprising: a battery assembly, an atomizer assembly, a cigarette-solution storage area, and a hollow shell having a mouthpiece: the battery assembly connects with the atomizer assembly, and both are located in the shell; the cigarette solution storage area is located in one end of the shell adjacent to the mouthpiece, and fits with at least a portion of the said atomizer assembly inside it; the shell has through-air-inlets; the atomizer assembly includes an atomizer comprising an electric heating rod and a run-through atomizing chamber; the electric heating rod comprises a cylinder and a heating element provided at the wall of the cylinder, the electric heating rod is in the said atomizing chamber and there is a negative pressure cavity in the atomizing chamber.
U.S. Patent No. 8,135,122 owned by NobelBiz Inc. Claim 1: functional language present
A system for processing an outbound call from a call originator to a call target, the system comprising: a database storing a plurality of outgoing telephone numbers; **an information processor** controlled by the call originator and **configured to** **process a trigger comprising a telephone number of the call target;** **access the database and select a replacement telephone number from the plurality of outgoing telephone numbers based on at least an area code of the telephone number of the call target;** **modify caller identification data of the call originator to the selected replacement telephone number,** the selected replacement telephone number having at least an area code the same as an area code of the telephone number of the call target; and **transmit the modified caller identification data of the call originator to the call target.**

U.S. Patent No. 5,560,360 owned by Neurografix Claim 1: functional language present
A method of utilizing magnetic resonance to determine the shape and position of mammal tissue, said method including the steps of: (a) exposing an in vivo region of a subject to a magnetic polarizing field, the in vivo region including non-neural tissue and a nerve, the nerve being a member of the group consisting of peripheral nerves, cranial nerves numbers three through twelve, and autonomic nerves; (b) exposing the in vivo region to an electromagnetic excitation field; (c) sensing a resonant response of the in vivo region to the polarizing and excitation fields and producing an output indicative of the resonant response; **(d) controlling the performance of the steps (a), (b), and (c) to enhance, in the output produced, the selectivity of said nerve,** while the nerve is living in the in vivo region of the subject; and **(e) processing the output to generate a data set describing the shape and position of said nerve, said data set distinguishing said nerve from non-neural tissue,** in the in vivo region to provide a conspicuity of the nerve that is at least 1.1 times that of the non-neural tissue, without the use of neural contrast agents, **said processing including the step of analyzing said output for information representative of fascicles found in peripheral nerves, cranial nerves numbers three through twelve, and autonomic nerves.**
U.S. Patent No. 6,462,713 owned by Transdata Inc. Claim 1: NO functional language
For use with an electric meter chassis having a dielectric housing protruding therefrom, an antenna for allowing electric meter circuitry located in a circuit board rack within said chassis to communicate wirelessly through said dielectric housing, comprising: a wireless communication circuit couplable to said electric meter circuitry; and an antenna element located within said dielectric housing proximate said circuit board rack, said antenna element coupled to said wireless communication circuit.
U.S. Patent No. 7,627,975 owned by Prototype Productions Inc. Claim 1: NO functional language
A firearm system comprising: a handguard power coupler comprising a handguard power input and at least one power connection; a handguard comprising at least one powered mounting rail comprising at least one rail power connection; wherein a power source electrically connected to the handguard power input is also electrically connected to the at least one rail power connection; and wherein a rail accessory attached to the at least one mounting rail receives electrical power from the power source.

U.S. Patent No. 7,742,084 owned by Eastman Kodak Co. Claim 1: functional language present
An electronic camera comprising: a communication interface; **a sensor for capturing images;** **a first memory for storing images;** **a second memory for storing information for communicating with a plurality of destinations via the communications interface;** and **a user interface for selecting an image destination and for commanding the camera to send the images to the selected destination via the communications interface using the information,** and wherein **the user interface displays a plurality of descriptive icons representative of the plurality of destinations** and selection is made by reference to at least one of the icons.
U.S. Patent No. 8,088,480 owned by Shieldmark Inc. Claim 1: NO functional language
An adhesive tape comprising: a polymer having a Shore A Hardness of between 92 and 100; and a layer of pressure sensitive adhesive comprising a first side and an opposed second side, the first side being in direct and uninterrupted contact with the polymer layer where the adhesive tape comprises an average thickness between 65 mil and 69 mil.
U.S. Patent No. 6,763,998 owned by United Coin Machine Co. Claim 1: functional language present
A system for securely storing and controlling the dispensing of a payout, comprising: **a payout dispenser, for securely storing and dispensing a payout;** **a terminal, for controlling the dispensing of the payout, adapted to be connected to the payout dispenser and to interface with a payout authorizing attendant in connection with the payout;** and a network for interconnecting the payout dispenser and the terminal.

U.S. Patent No. 7,931,199 owned by Serverside Group Limited Claim 1: functional language present
Computerized financial transaction card production equipment operable to apply one or more personalized images to a financial transaction card, the production equipment comprising: **a module configured to receive a personalized image of a customer,** the image being received from **an image processor computer arranged to facilitate image personalization by remote customers;** a module configured to receive a customer identifier that corresponds to the remote customer that personalized said image; a module configured to receive a financial record of the remote customer that personalized the image; **a card printer arranged to print images on card material and equipment configured to apply financial information from the financial record to the card material;** and **a controller operable,** based on said customer identifier, **to cause printing of said personalized customer image onto the card material and to cause application of relevant financial information from the financial record onto the card material,** wherein the customer identifier comprises an identifier selected from a secure unique identifier and a one-way code.
U.S. Patent No. 7,982,720 owned by Immersion Corp. Claim 1: functional language present
A haptic feedback device comprising: **a display configured to display one or more graphical items,** at least one of which has an active state; and **an actuator configured to impart to the haptic feedback device a haptic force** associated with a displayed graphical item that is in an active state and a second haptic force associated with a displayed graphical item that is in an inactive state.
U.S. Patent No. 6,293,556 owned by Krausz Industries Claim 1: NO functional language
A sealing ring for pipe connector means made of resilient material, the sealing ring comprising a first sleeve-like ring the cross section of which defines a inner space therein, and a second ring overriding said first sleeve-like ring and being loosely connected to said first ring, said second ring being adapted to be torn off said first ring at a predetermined location so as to adapt the sealing ring to interconnect pipes of substantially different diameters.
U.S. Patent No. 6,722,686 owned by Cequent Performance Products Inc. Claim 1: functional language present
A device for closing the a socket of an unhitched trailer hitch coupler, said device comprising: a locking bar wherein at least a portion thereof rests on top of the trailer hitch coupler; and a base comprising a plug member for receipt within said trailer hitch coupler socket, a locking bar-receiving aperture, and an integral **locking means for lockingly engaging said locking bar within said aperture.**

U.S. Patent No. 6,400,376 owned by Ericsson Inc. Claim 1: functional language present
In a data storage device including a screen portion for visually displaying a part of a virtual page larger than said screen portion whereby only a portion of the virtual page is displayed in said screen portion, a display control structure comprising: at least **one sensor** mounted on the device and **configured to sense changes in position of the device in a reference coordinate system** and transmit signals indicative of said changes; **a control circuit adapted to pan said virtual page over said screen portion responsive to signals from said sensor indicative of said position changes when said control circuit is in a panning mode;** at least one touch-responsive first area on said screen portion, said **first area when touched by a user placing said control circuit in said panning mode;** and a touch-responsive second area on said screen portion, said **second area when touched by a user placing said control circuit out of said panning mode** and said second area being the part of the virtual page displayed on said screen portion when said device is in said panning mode.
U.S. Patent No. 8,071,577 owned by Bayer Pharma AG Claim 1: NO functional language
A multiphase product for contraception comprising: a first phase of 2 daily dosage units, each comprising 3 mg of estradiol valerate, a second phase of 2 groups of daily dosage units, a first group comprising 5 daily dosage units, each of which comprises 2 mg of estradiol valerate and 2 mg of dienogest, and a second group comprising 17 daily dosage units, each of which comprises 2 mg of estradiol valerate and 3 mg of dienogest; a third phase of 2 two daily dosage units, each comprising 1 mg of estradiol valerate, and a fourth phase of 2 two daily dosage units, each comprising a pharmaceutically acceptable placebo.
U.S. Patent No. 5,949,880 owned by Maxim Integrated Products Inc. Claim 1: functional language present
A method for electronically transferring units of exchange between a first module and a second module, comprising the steps of: **a. initiating communication between said first module and an electronic device;** b. passing a first value datum from said first module to said electronic device; c. passing said first value datum from said electronic device to said second module; **d. performing a mathematical calculation on said first value datum thereby creating a second value datum;** e. passing said second value datum from said second module to said electronic device; f. passing said second value datum from said electronic device to said first module; g. storing said second value datum in said first module; and h. discontinuing communication between said first module and said electronic device.

U.S. Patent No. 7,139,974 owned by Bascom Research LLC Claim 1: functional language present
A method for providing a framework for document objects located on a network, the method comprising: **creating one or more link directories** for storing link relationships between document objects located on the network, wherein the one or more link relationships are separate from the document objects; **creating a link relationship between a first document object located on the network and a second document object located on the network;** **assigning attributes describing the link relationship,** wherein the attributes or references to the attributes are stored with the link relationship; **using a unique identifier to retrieve the link relationships,** wherein unique identifiers indicate locations of document objects on the network; and **presenting the link relationship with one or more of the attributes describing the link relationship.**

Presentation to the 2/12/13 Software PTO Roundtable at SLS. Slides 39 &40 updated (strikethrough).

Santa Clara University

Software Patents & Functional Claiming

Assistant Professor Colleen Chien & Aashish Karkhanis, SCU Law '13

colleenchien@gmail.com

Santa Clara University

Why should we care about software patents?

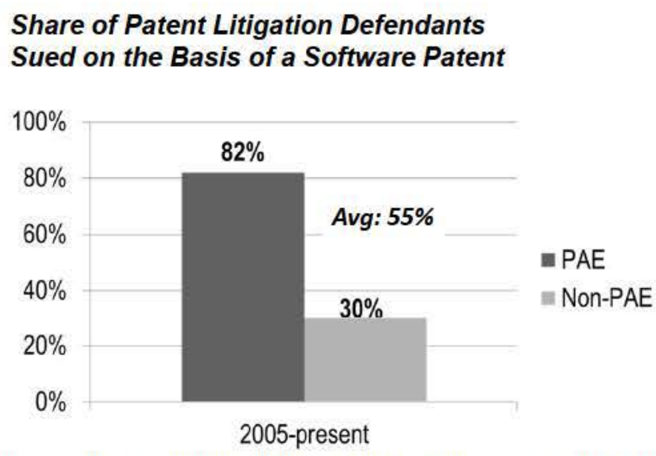
Software patents have attracted a disproportionate amount of attention about the patent system

Software patents have attracted a disproportionate amount of attention about the patent system

Is the attention on software patents warranted?

Yes. Software patents are behind a disproportionate share of patent disputes

As many as 55% of all patent defendants and 82% of PAE ("patent troll" defendants have been sued on the basis of a software patent



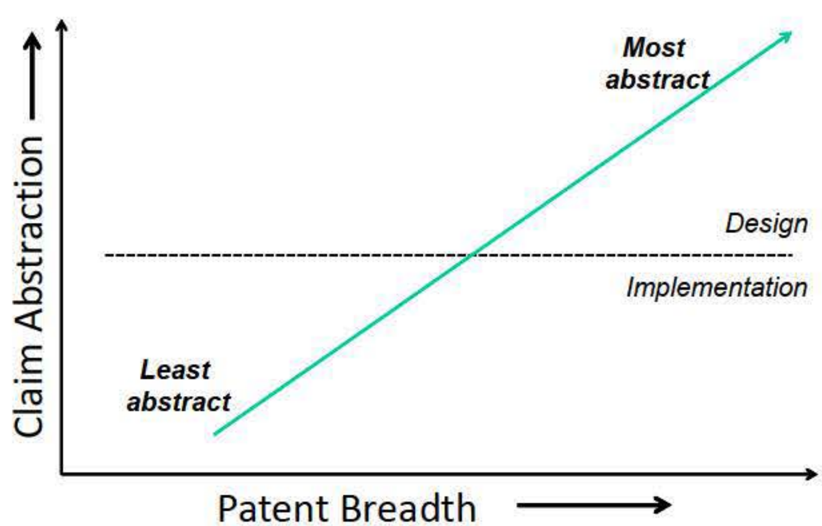
RPX RATIONAL PATENT

Gazelle

Class-based definition of "software" patent: Graham & Vishnubhakat, Journal of Ec. Perspectives. 27:1 (2013) which notes that this definition may contain false positives and negatives. Based on an analysis by Gazelletech of data provided by RPX Corp. © 2012-current suit #s: 86%/35% PAE/non-PAE, respectively.

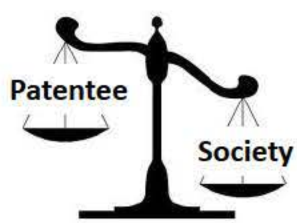
Software patents have disproportionately been asserted by PAEs (patent "trolls"). WHY?

Software is abstract. The more abstractly a patent is claimed, the larger its footprint on others



There is a perception that "bad" software patents are breaking the patent system

There is a perception that "bad" software patents are breaking the patent system



But "bad" software patents are difficult to weed out
By many measures, PTO examination is just as rigorous of software patents as of non-software (Graham & Vishnubhakat)
Patentable subject matter (101) line-drawing is difficult, impossible?
Novelty and nonobviousness screens (102/103) are costly to apply

Today: If those levers aren't working how about 112 (the disclosure doctrines)?

Why don't we more forcefully apply the disclosure law (35 USC 112(b) and 35 USC 112(f)) to rebalance the patent bargain <u>without changing the patent statute?</u>

This Presentation tests the premise that greater application of 112(f) would help. How?

112 (f)

x

PAE Patents

Lodsys, LLC

.
.
.

+

Technical abstraction framework

We examine how well-supported functionally claimed PAE patents are. Are they "crap"? Or are they actually well-supported?

What we did

112 (f)

X

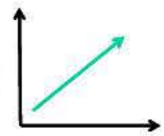
PAE Patents

Lodsys, LLC

.
.
.

+

Technical abstraction framework



1. **Develop ways to identify functional claims**
2. **Apply to PAE and non-PAE patents**
3. **Look for support for functionally claimed PAE patents**

Our analysis creds

Aashish R. Karkhanis Reg. # 62,572
SCU Law '13

B.S., Computer Engineering,
Virginia Tech

Patent Prosecutor, 4 Years
Patent Examiner, 2 Years (AU 3714)

Colleen V. Chien Reg. # 55,062

B.S., Engineering
A.B., Science Technology & Society, Stanford

Full-Time IP Litigator and Patent Prosecutor,
4 Years, Fenwick & West

What we did

Step 1: identify functionally claimed patents

112 (f)

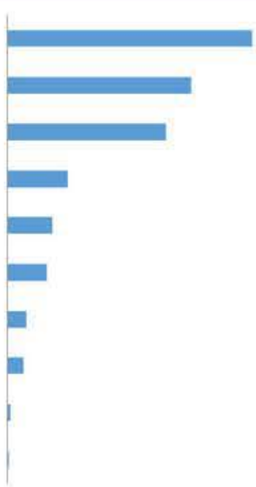
Key words/phrases
[see, e.g. Lemley 2013 & MPEP]

"configured to", "permitting…", "programmable means for," "capable of engaging," "adapted to," "for…ing," "operable to…", "mechanism", "data processing system"
"mechanism for," "module for," "device for," "unit for," "component for," "element for," "member for," "apparatus for," "machine for," or "system for."

Thanks to Bob Hulse (Partner, Fenwick & West) for help with method based (step + function) claiming

Step 2: Apply it to PAE and non-PAE litigated patents

The Patent Freedom Dataset – 10 PAE litigated patents, 1 each selected from the following campaigns



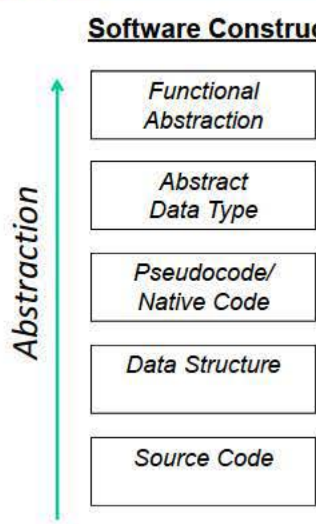
	Defendants*	Lawsuits	Patents	Technology
GeoTag Inc	435	115	1	Associating online information with geographic areas
ArrivalStar	326	211	16	Vehicle tracking and notification
PJC Logistics LLC	281	44	1	Vehicle tracking and monitoring
Lodsys LLC	106	36	4	Customer-based product design module
Blue Spike LLC	79	56	4	Digital fingerprinting
Datatern Inc	70	28	2	Interfacing object oriented software applications with relational database
Ogma LLC	32	10	1	Programmable motion-sensitive sound effects device
Kelora Systems LLC	27	11	1	Guided parametric search and retrieval
Project Paperless LLC	3	3	2	Distributed electronic document management
Single Touch Systems Inc	1	1	1	Management and administration of media streaming

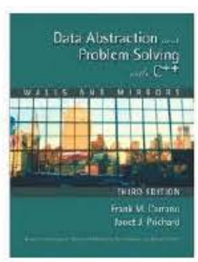
*Includes a small number of DJ cases where the operating company is a plaintiff

The Patent Freedom Dataset – control group of 20 non-PAE litigated patents

Half highly litigated, half randomly selected
Submission will include details

Step 3: Evaluate per a textbook technical abstraction framework





Carrano and Prichard, Chapter 3: "Data Abstraction, the Walls"

Case Study Examples – 5 litigated PAE patents

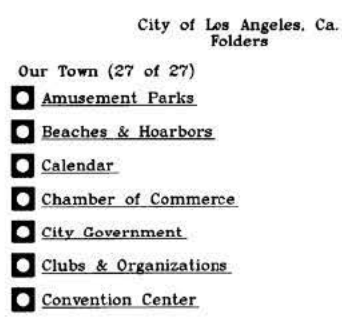
Step 3: Evaluation per a textbook technical abstraction framework

Abstraction

Software Construct	Definition
Functional Abstraction	*Conceptually, what the software program will do.*
Abstract Data Type	*A collection of data and set of operations on them.*
Pseudocode/ Native Code	*A set of instructions that specifies the operations that collectively achieve the function.*
Data Structure	*A programming language construct that stores a collection of data.*
Source Code	*Human-readable computer code before it is compiled into machine readable object code.*

Case Study Examples – 5 litigated PAE patents

"Geolocation/ Where's the closest Starbucks?" U.S. 5,930,474 Asserted by GeoTag



435	115	1	45
defendants	lawsuits	patent	pages

The '474 Patent, Distilled

delivering info "such as business services, entertainment, news, consumer goods" for a user's local area

See U.S. Patent No. 5,930,474 at col. 9, lines 28-35.

Functional Abstraction in '474

"… if a user is interested in finding an out-of-print book, or a good price on his favorite bottle of wine, but does not want to travel outside of the Los Angeles area to acquire these goods, then **the user can simply designate the Los Angeles area as a geographic location for which a topical search is to be performed** … the geographic topical organization format provided in accordance with the preferred embodiment provides the user with a valuable Internet organizing tool"

U.S. Patent No. 5,930,474 at col. 7, lines 5-29.

Abstract Data Types in '474

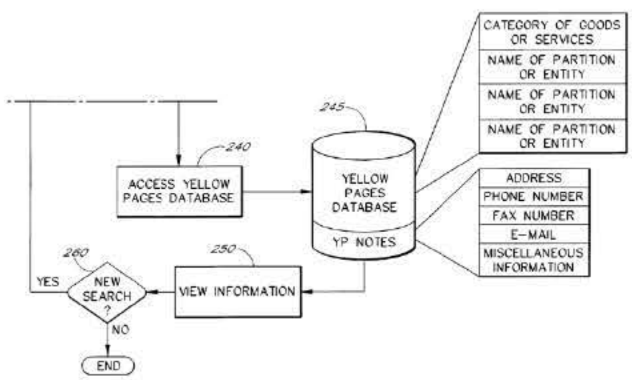


FIG. 2C

U.S. Patent No. 5,930,474 at fig. 2C.

Pseudocode in '474

"This parameter may be used by the **Read subroutine** 320 whenever there are more than 50 entries in a list and scrolling is to be supported. In a preferred embodiment, the **first search has this value always entered as zero, and subsequent scroll searches increment this value to support scrolling**. Finally, the NameKey parameter indicates the name of the folder to display … Any **entry whose parent folder name matches the name specified will be returned** by the search."

U.S. Patent No. 5,930,474 at col. 12, lines 35-45.

Data Structures in '474

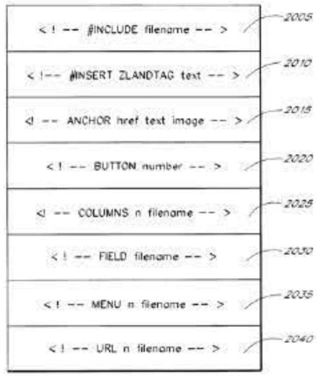


FIG. 20

U.S. Patent No. 5,930,474 at fig. 20.

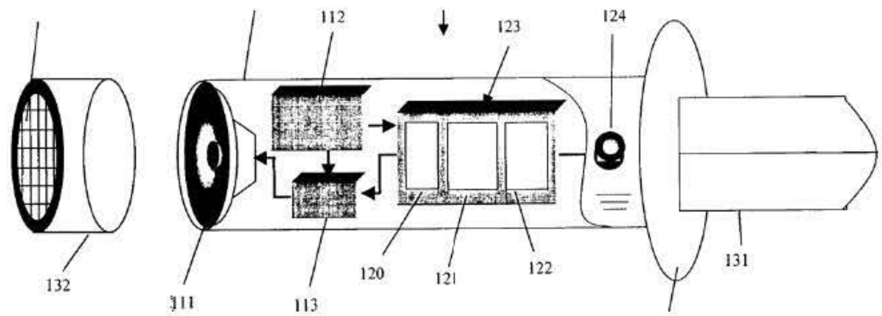
Source Code in '474

TABLE 3

```
Content-type: text/html
<HTML><HEAD><TITLE>City of Irvine, CA:City1.6.3</TITLE></HEAD><body background="/gifs
/rock1.gif"><center><B>City of Irvine, CA</B> <br><B>Folders</B></center><br><DL><DT><IMG
SRC="/icons/folder_open.gif" hspace=10 border=0> <B>School Listing</B> (3 of 3) <DD><IMG
SRC=/Bullets/yellow.gif hspace=10 border=0 ALT=" "> <a HREF="/search/yp2?na\us\ca\
sc\ir\cityyp+STDYP+KeywordListing+0+Colleges_&_Universities"> Colleges & Universities</a>
<DD><IMG SRC=/Bullets/yellow.gif hspace=10 border=0 ALT=" "> <a HREF="/search/yp2?na\us\ca\
sc\ir\cityyp+STDYP+KeywordListing+0+Preschool_&_Kindergarten"> Preschool & Kindergarten</a>
<DD><IMG SRC=/Bullets/yellow.gif hspace=10 border=0 ALT=" "> <a HREF="/search/yp2?na\us\ca\
sc\ir\cityyp+STDYP+KeywordListing+0+Secondary_&_Elementary">Secondary & Elementary </a><br>
</DL> <p><a href="/guides/default.htm"> <img src="/gifs/guides.gif" align=middle hspace=10
border=0 alt="Guided Tours"></a> <a href="/default.htm"> <img src="/gifs/zbutton.gif"
align=middle hspace=10 border=0 alt="Z Land"></a></p> <IMG SRC="/Lines/eyes.gif" alt=" ">
<p>Send your letters and comments to: <a href="mailto:webeditor@mail.zland.com">webeditor@
mail.zland.com</a><br><br></p> Copyright &copy; 1995 Z Land, LLC. All rights reserved.
</BODY>
</HTML>
```

U.S. Patent No. 5,930,474 at cols. 27-28.

"The Light Saber"
U.S. 6,150,947 Asserted by Ogma



32	10	1	14
defendants	lawsuits	patent	pages

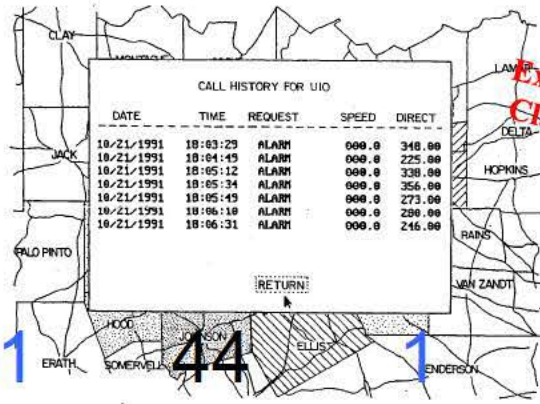
Object Code in '947 (Ogma)

TABLE 1

S22500000040060093FE2040020093FE1040000093FE3040087093FE403C008C0000003C0083EF
S22500002102800018020F000000000000000000000000000000000000000000000000000000000F
S22500004200000000000000000000000000000000000A001F0000000000000000000000401F50C0
S22500006393FE2040004093FE1040001093FE3040019093FE40028000000000000000000000070
S225000084000000000000000000000000000000000000000000000000000000000000000056
S2250000A5000000000000000000000000000000000000000000000000000000000000000035
S2250000C6000000000000000000000000000000000000000000000000000000000000000014
S2250000E70000000000000000000000000000000000000000000000000000000000000000F3
S2250001080000000000000000000000000000000000000000000000000000000000000000D1
S22500012900000000000000000000000000000040227093FE2040000093FE1040008093FE30CB
S22500014A4022E093FE40028000000000000000000000000000000000000000000000000000FA
S22500016B00000000000000000000000000000000000000000000000000000000000000006E
S22500018C00000000000000000000000000000000000000000000000000000000000000004D
S2250001AD00000000000000000000000000000000000000000000000000000000000000002C
S2250001CE00000000000000000000000000000000000000000000000000000000000000000B
S2250001EF000000000000C00000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC000C1BB
S22500021000C288C388C488C588C680C780C853C909CA00CC40CD0018030F0A001F0A001F0AF7
S225000231001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0AE4
S225000252001F0A001F80003A227A0F0A001018078F180B0F0A001F0A001F0A001F18148F0A9F
S225000273001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0AA2
S225000294001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A81
S2250002B5001F40800093FFF034000234003A34004334003B3801013800D902020F47B0009350
S2250002D6FE60000000000000000000000003400163800154072B093FF3040000093FF404097
S2250002F7000093FF504860F093FF6040007093FF7040007093FF9040007093FF9040007093EC
S225000318FFA040000093FFE041800093FFF03C0FFC0000003C00243C0401400010900030042F
S22500033900603C040360000E0D0C9080003026780F1805F14000248000023800F1806308015
S22500035A000023800F1806610280004BF3F480016023800F90004A028000800170238 00F9076

U.S. Patent No. 6,150,947 at col. 7, lines 17-25; cols. 9-10.

"Is my Train on Time?" U.S. 5,223,844 Asserted by PJC Logistics



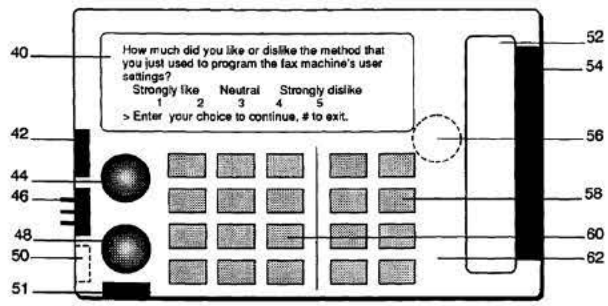
281 defendants | 44 lawsuits | 1 patent | 41 pages

Functional Abstraction in '844 (PJC Logistics)

"In a preferred embodiment mapping 182A displays a general area coverage map a relatively large area, such as the 14 counties around the Dallas/Fort Worth metroplex area. **Mapping displays 182B, 182C, and 182D may be used display vehicle locations for both stolen vehicle and motorist assistance calls on much smaller maps**."

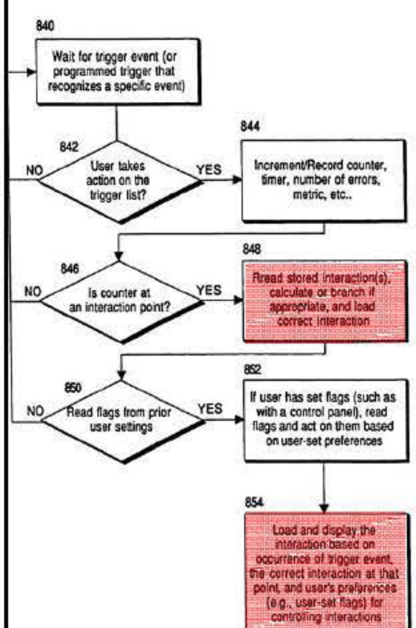
U.S. Patent No. 5,223,844 at col. 27, lines 27-32.

"User Feedback Interface"
U.S. 7,222,078 Asserted by Lodsys



106	36	4	89
defendants	lawsuits	patents	pages

FIGURE 23



Functional Abstraction in '078 (Lodsys)

Pseudocode Describing Handling of Variables

Functional Description of User Interaction

U.S. Patent No. 7,222,078 at fig. 23.

Functional Description of User Interaction Preferences

"Printing e-Documents"
U.S. 6,185,590 Asserted by Project Paperless

3	3	2	40
defendants	lawsuits	patents	pages

Functional Abstraction in '590 (Project Paperless)

"Loading and unloading the engine (DLLs provided into and out of memory)*

Mapping original functions to engine object counterparts

Adding general error detection and correction*

Determining and matching arguments and return values for mapping the original functions to their engine object counterparts In order to add assertion and error detection and correction, the original function must be wrapped and called from within the engine object version of the original function.

Managing error feedback. All APIs have their own way providing error feedback. Since one of the goals of the Engine Management layer is to generically manage error detection, correction, and feedback, it must handle all errors identically … **By creating specific classes of APIs the process of generating Layer 1 engine management may be expedited manually and/or automatically.**"

U.S. Patent No. 6,185,590 at col. 17, lines 29-50.
*source code disclosed: U.S. Patent No. 6,185,590 at cols. 15-16.

Our findings: all 10 PAE patents were functionally claimed, but the supporting disclosure varied

We found (N=30):

PAE litigated patents were always functionally claimed (100%), but functional claiming was also prevalent among non-PAE litigated patents (50%)

Among the 10 PAE patents, the supporting disclosure varied significantly, ~~40% of the patents contained only functional abstraction, but the other 60% contained more, e.g. pseudocode and ADT type disclosure~~ 70% contained claims that had one or more unsupported (disclosure only at the functional abstraction level) elements, others had more.

"Not all code is created equal" the contribution conferred via pseudo or source code varied. Source code over generic steps didn't add much.

Implications

Does functional claiming correctly identify the problem?

Yes but may be overinclusive? Applies to non-s/w patents too. Narrow to PoN FC?

What is the payoff for construing more claims as 112(f)?

Existing patent claims and applications likely to be invalidated – ~~4~~70% of studied high impact patents included claims that had one or more unsupported elements~~didn't include more than functional abstraction~~. Others will be narrowed in scope.

How should supported claims be construed?

Need clarity around this to avoid creating even more uncertainty. What are equivalents of ADT, pseudocode, source code?

What would heightened application of 112(f) do to filing incentives?

Better disclosure. Delayed application.

Recommendation: if guidelines, phased introduction of them to allow prosecutors time to change their practices.

Thank you!