



2-12-2013

Software Patents & Functional Claiming

Colleen Chien

Santa Clara University School of Law, colleenchien@gmail.com

Aashish Karkhanis

Follow this and additional works at: <http://digitalcommons.law.scu.edu/facpubs>



Part of the [Intellectual Property Commons](#)

Automated Citation

Colleen Chien and Aashish Karkhanis, *Software Patents & Functional Claiming* (2013),
Available at: <http://digitalcommons.law.scu.edu/facpubs/603>

This Conference Paper is brought to you for free and open access by the Faculty Scholarship at Santa Clara Law Digital Commons. It has been accepted for inclusion in Faculty Publications by an authorized administrator of Santa Clara Law Digital Commons. For more information, please contact sculawlibrarian@gmail.com.

Santa Clara
University

Presentation to the
2/12/13 Software PTO
Roundtable at SLS

Software Patents & Functional Claiming

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

colleenchien@gmail.com



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

hulu™

Lodsys, LLC

+Colleen Search Images Maps Play YouTube

Google

- software patents
- software patents
- software patents **debate**
- software patents **pros and cons**
- software patents **are evil**

How Valuable is
Amazon's 1-Click



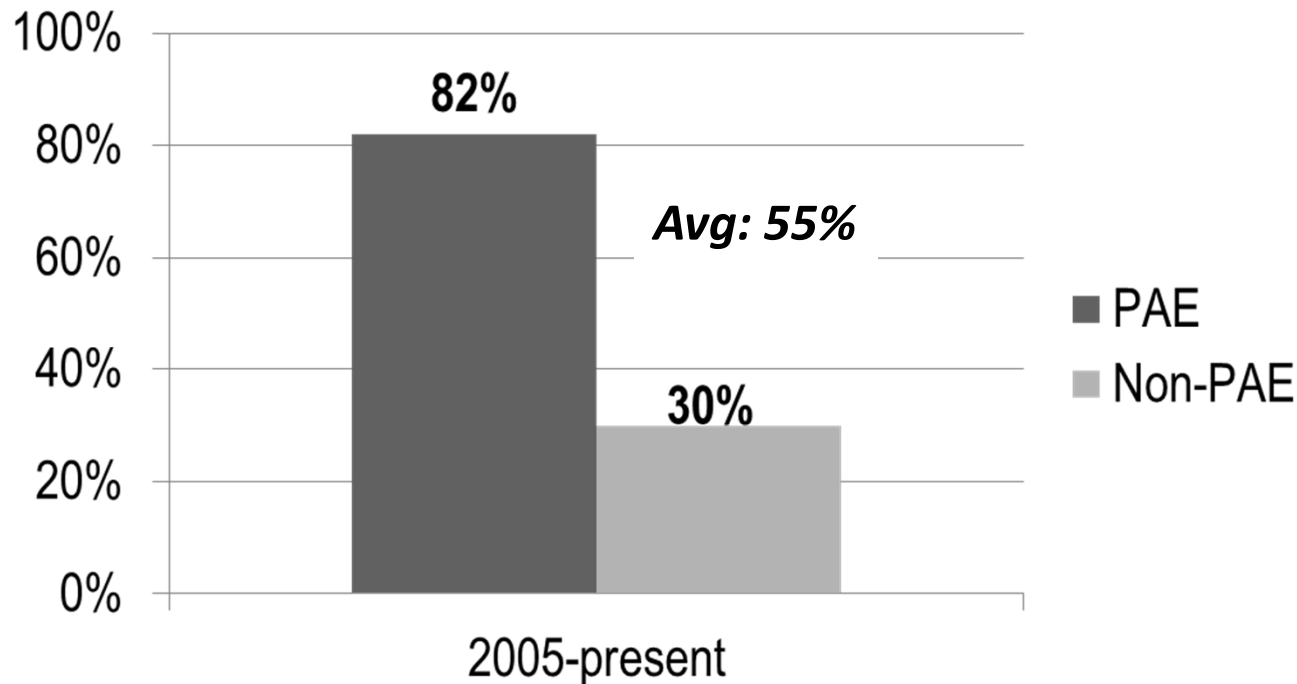
~~project
paperless~~

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

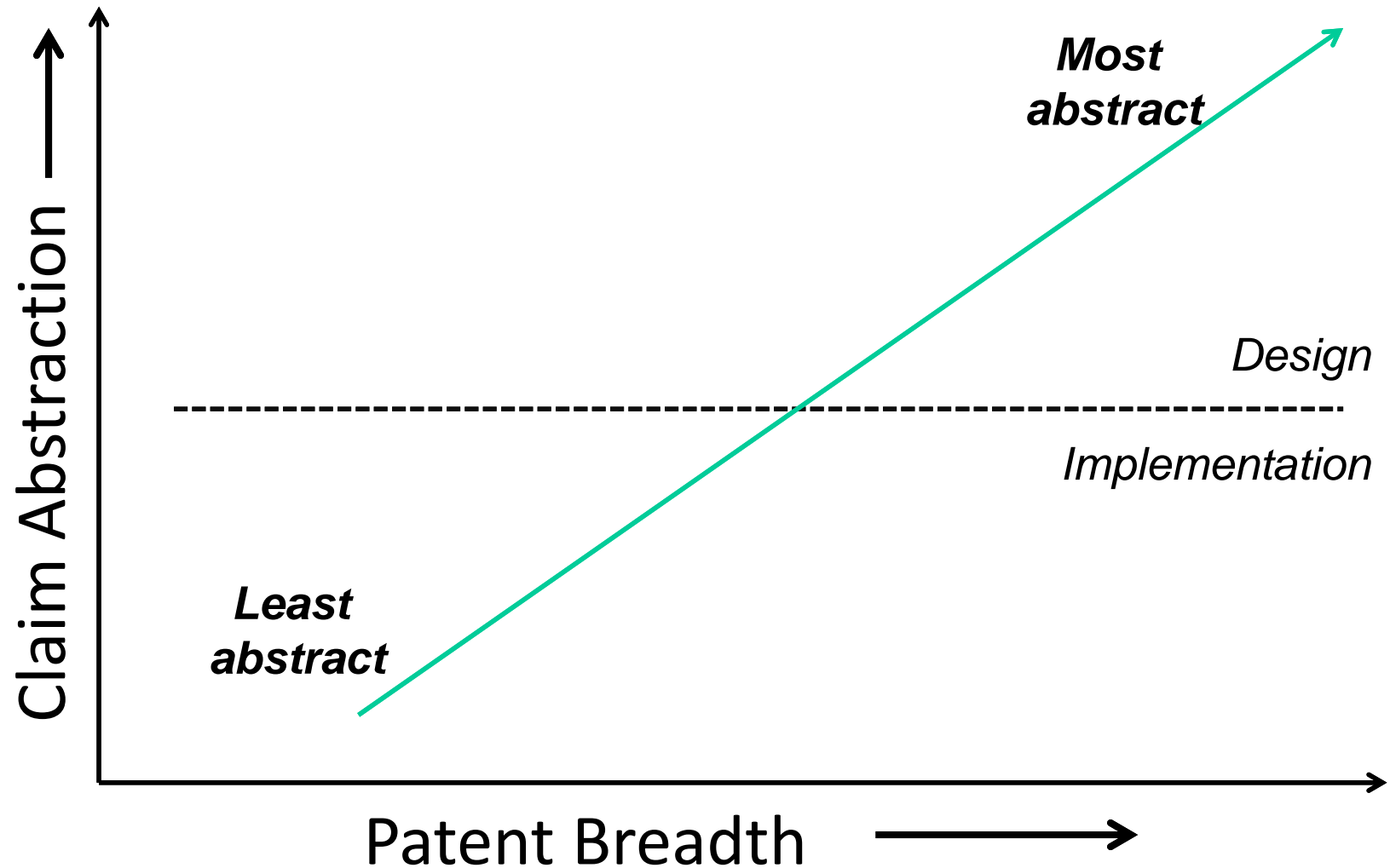
Share of Patent Litigation Defendants Sued on the Basis of a Software Patent



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.

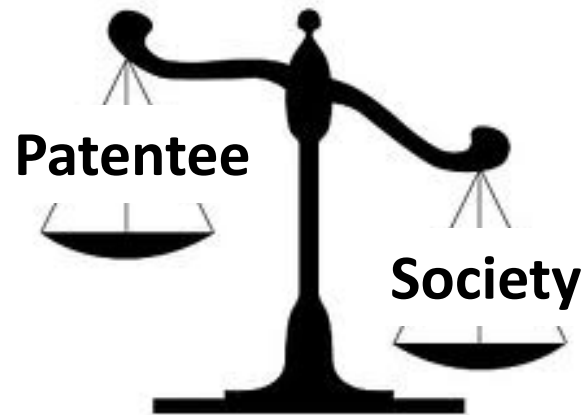
Software patents have disproportionately been asserted by PAEs (patent “trolls”). WHY?

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 without changing the patent statute?**

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

112 (f)



PAE Patents

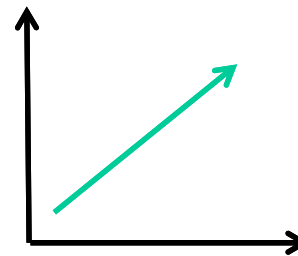


X

.
. .
. . .

+

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)



PAE Patents

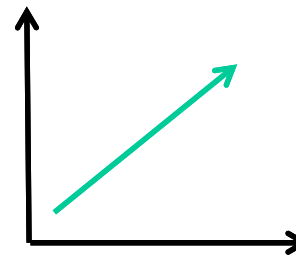
Lodsys, LLC

x

.
. .
. .

+

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

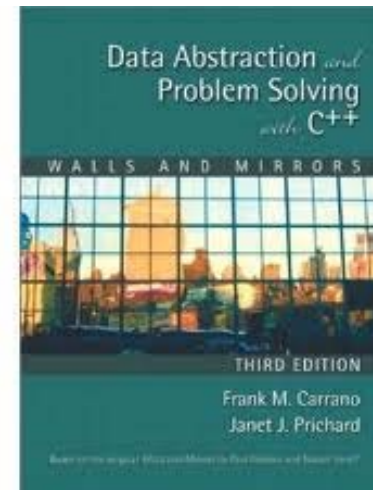
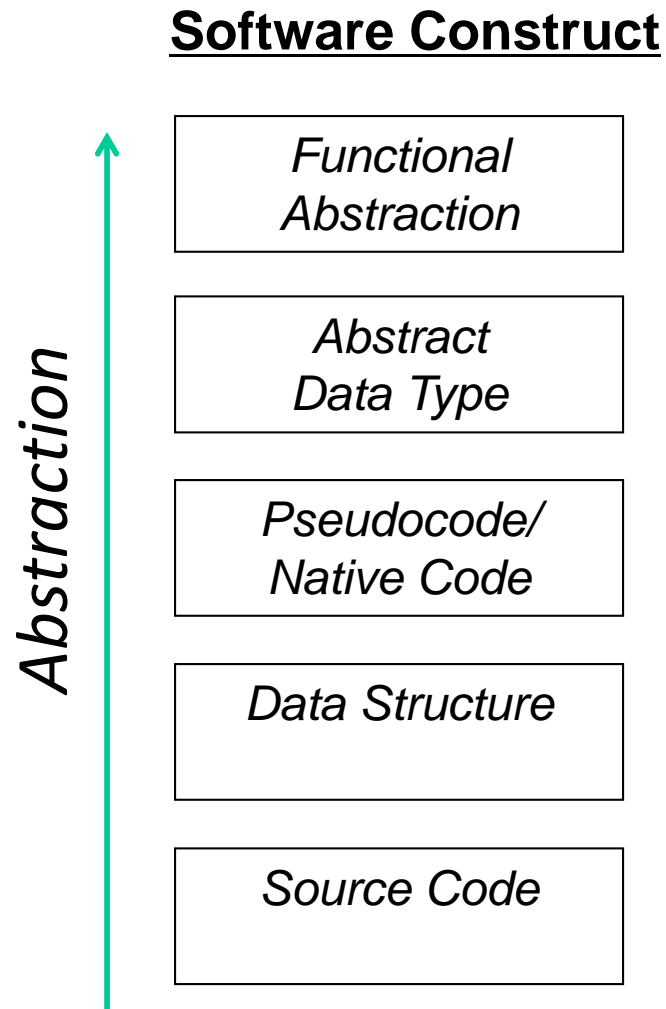
	Defendants*	Lawsuits	Patents	Technology
GeoTag Inc	435	115	1	Associating online information with geographic areas
ArrivalStar	326	211	16	Vehicle tracking and notification
PJC Logistics LLC	281	44	1	Vehicle tracking and monitoring
Lodsys LLC	106	36	4	Customer-based product design module
Blue Spike LLC	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

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

Case Study Examples – 5 litigated PAE patents

“Geolocation/ Where’s the closest Starbucks?” U.S. 5,930,474 Asserted by GeoTag

City of Los Angeles, Ca.
Folders

Our Town (27 of 27)

- Amusement Parks
- Beaches & Hoarbors
- Calendar
- Chamber of Commerce
- City Government
- Clubs & Organizations
- Convention Center

435

defendants

115

lawsuits

1

patent

45

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

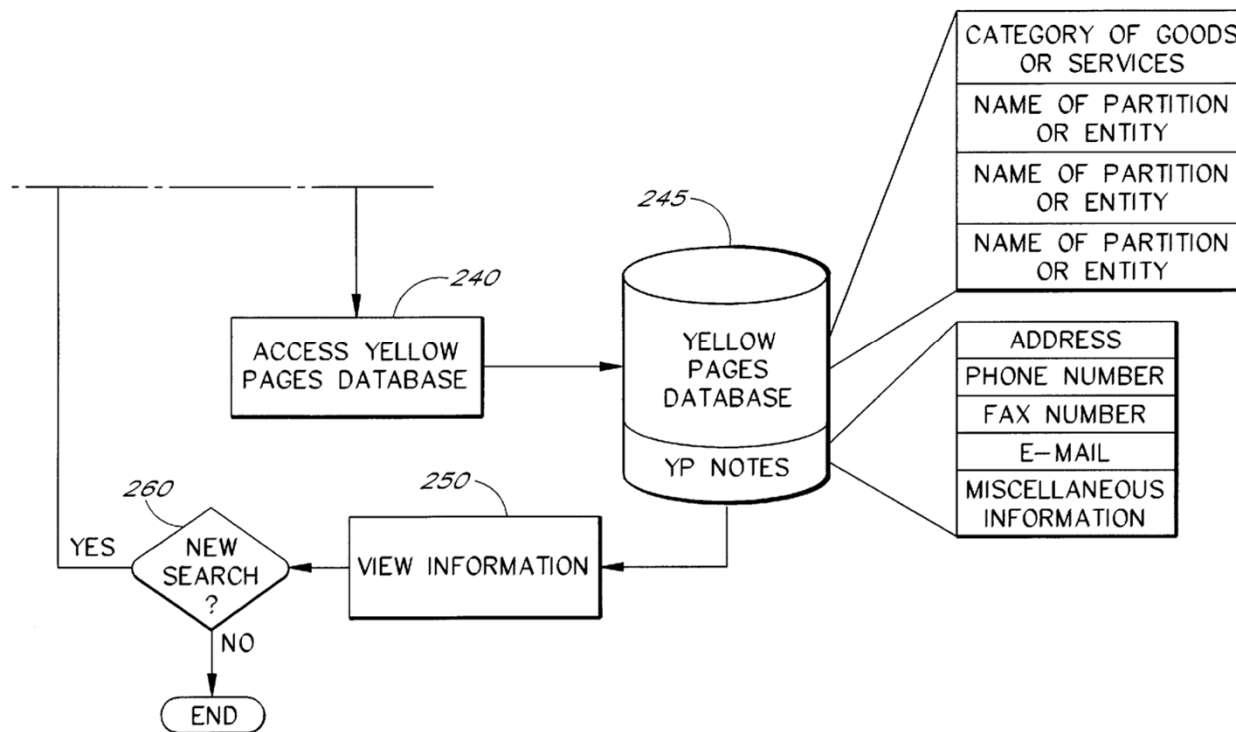


FIG. 2C

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

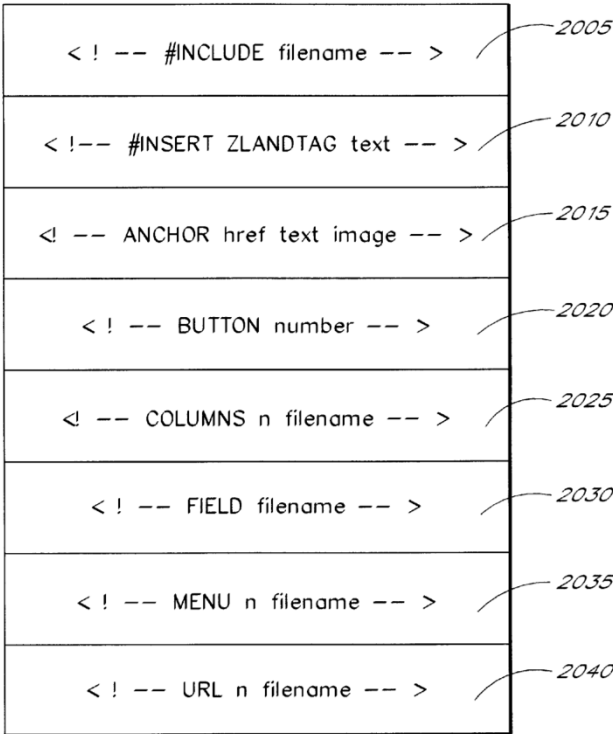


FIG. 20

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

Source Code in '474

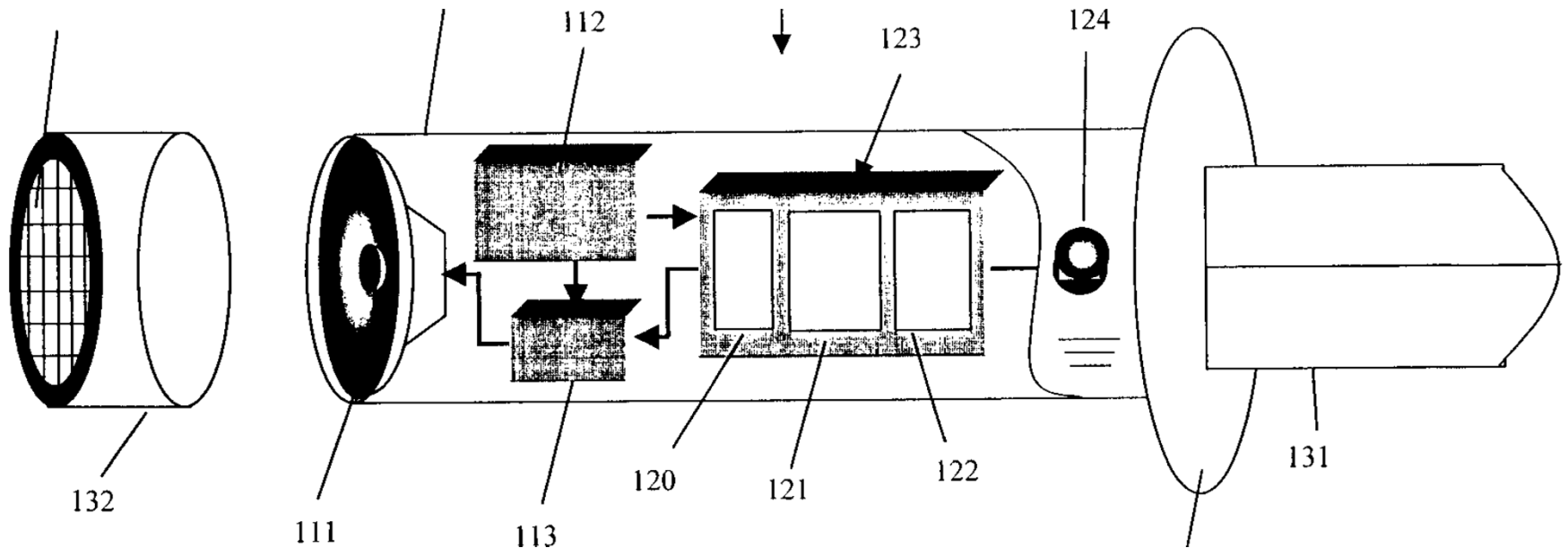
TABLE 3

```
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\
se\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\
se\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\
se\ir\cityyp+STDYP+KeywordListing+0+Secondary_&_Elementary">Secondary & Elementary </a><br>
</DL> <p><a href="/guides/default.htm"> </a> <a href="/default.htm"> </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

defendants

10

lawsuits

1

patent

14

pages

Object Code in '947 (Ogma)

TABLE 1

S2250000040060093FE2040020093FE1040000093FE3040087093FE403C008C0000003C0083EF
S22500002102800018020F000F
S225000042000F
S22500006393FE2040004093FE1040001093FE3040019093FE4002800000000000000000000070
S22500008400056
S2250000A50035
S2250000C60014
S2250000E700F3
S225000108000D1
S22500012900D1
S22500014A4022E093FE4002800FA
S22500016B0006E
S22500018C004D
S2250001AD002C
S2250001CE00B
S2250001EF00000000000000C00000000000FFFFFFF000000000000000000000000000000000000C1BB
S22500021000C288C388C488C588C680C780C853C909CA00CC40CD0018030F0A001F0A001F0AF7
S225000231001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0AE4
S225000252001F0A001F80003A227A0F0A001018078F180B0F0A001F0A001F0A001F18148F0A9F
S225000273001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0AA2
S225000294001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A001F0A81
S2250002B5001F40800093FFF034000234003A34004334003B3801013800D902020F47B0009350
S2250002D6FE6000000000000000000000000000003400163800154072B093FF304000093FF404097
S2250002F7000093FF504860F093FF6040007093FF7040007093FF9040007093FF9040007093EC
S225000318FFA040000093FFE041800093FFF03C0FFC0000003C00243C0401400010900030042F
S22500033900603C040360000E0D0C9080003026780F1805F140002480000023800F1806308015
S22500035A000023800F1806610280004BF3F480016023800F90004A02800080017023800F9076

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

DATE	TIME	REQUEST	SPEED	DIRECT
10/21/1991	18:03:29	ALARM	000.0	348.00
10/21/1991	18:04:49	ALARM	000.0	225.00
10/21/1991	18:05:12	ALARM	000.0	338.00
10/21/1991	18:05:34	ALARM	000.0	356.00
10/21/1991	18:05:49	ALARM	000.0	273.00
10/21/1991	18:06:10	ALARM	000.0	280.00
10/21/1991	18:06:31	ALARM	000.0	246.00

RETURN

*Ex Parte Reexam
Claims Amended*

281

44

1

41

defendants

lawsuits

patent

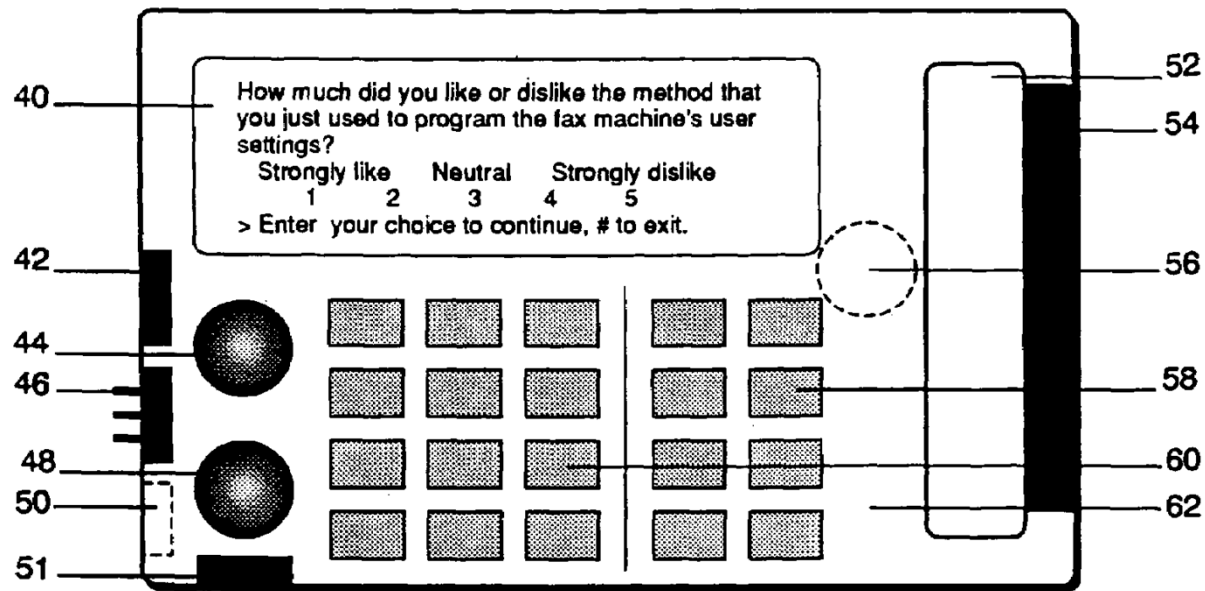
pages

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

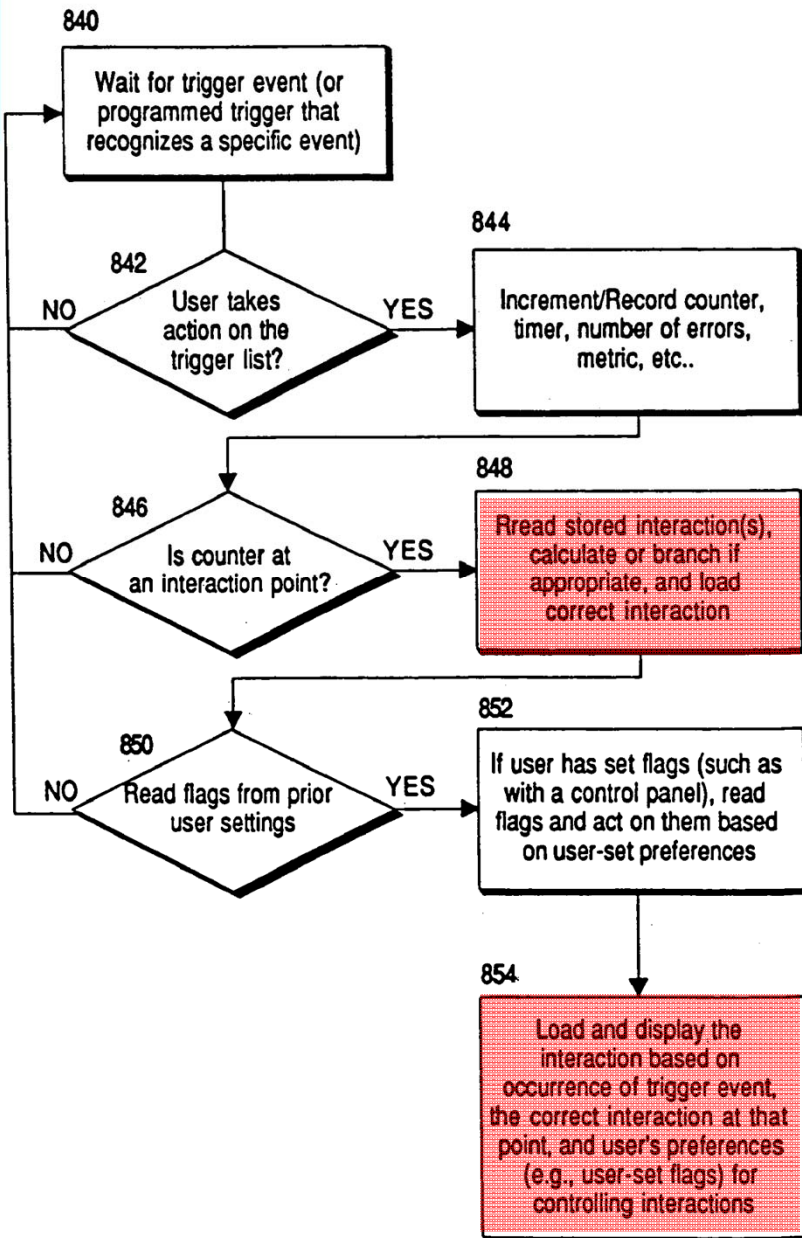
defendants

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

defendants

3

lawsuits

2

patents

40

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

“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 patents and applications likely to be invalidated – 40% of PAE patents 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!