# Functional Claiming

Lee Van Pelt
Van Pelt, Yi, and James

Thanks to Mark Lemley for providing the content of the case summary slides

# How Broadly Can you Claim Your Invention?

- One question, at least three ways to get at it.
- Scope of enablement
- Written description requirement
- 101 / Morse preemption of an abstract idea

# A brief history of functional claiming 1

- Before the early 90’s, means plus function was widely perceived to be the broadest way to claim.
- CAFC made it clear that it was not.
- Conventional strategy evolved to use functional language but not the term “means”
- In the mid 90’s, some defendants tried to identify functional claim language without “means” and narrow it under 112(6) but CAFC held a strong presumption that they were not m+f unless they used the term “means.”
- On the software side, the idea that software claims are step plus function claims never got any traction even if they said “step.”
- Expressed intent of the drafter usually held sway (except for software) (with exceptions).

# A brief history of functional claiming 2

- To preserve the drafter’s “intent,” courts would hold that functional language like ADC or charge pump or high voltage generator or propeller implied some specific structure.
- This was a golden era for patentees where courts were not applying 112(6) to narrow functional claims but not also invalidating them under Halliburton.
- Perception was that 112(6) “overruled” or “undid” Halliburton
- By 2000, 112(6) claims were considered to be rather old fashioned and by 2010 only a small percentage of claims contained m+f elements
- This led, among other things to some VERY broad software patents that claimed a result, not a way of achieving the result.

# The backlash

- 101. We would not need to shoehorn 101 into solving the problem of overbroad software functional claims if functional claiming was dealt with properly in software patents. See “SOFTWARE PATENTS AND THE RETURN OF FUNCTIONAL CLAIMING” by Lemley.
- If perceived intent is the test, then be careful what you intend because if you step out from under the 112f umbrella you could get nailed by Halliburton via 112(1) or 112(2).
- 112(6) was once perceived to make the PTO’s job harder. (The “other” holding in Alappat.)

# Functional Claiming and Point of Novelty 1

- Functional language is a problem at the point of novelty. Halliburton Oil Well Cementing Co. v. Walker, 329 U.S. 1 (1946)
- “A claim which describes the most crucial element in a "new" combination in terms of what it will do, rather than in terms of its own physical characteristics or its arrangement in the new combination, is invalid.”

# Functional Claiming and Point of Novelty 2

- For some claims, there is a polygon of novelty, but then the problem is claiming “any way of combining these functional elements”  (i.e. software “engine” claims)

# In re Hyatt

- Be careful not to write a single means claim
- MPEP 2164.08(a) Single Means Claim
  - single means claim, i.e., where a means recitation does not appear in combination with another recited element of means, is subject to an undue breadth rejection under 35 U.S.C. 112, first paragraph. *In re Hyatt*, 708 F.2d 712, 714-715, 218 USPQ 195, 197 (Fed. Cir. 1983) (A single means claim which covered every conceivable means for achieving the stated purpose was held nonenabling for the scope of the claim because the specification disclosed at most only those means known to the inventor.). When claims depend on a recited property, a fact situation comparable to *Hyatt* is possible, where the claim covers every conceivable structure (means) for achieving the stated property (result) while the specification discloses at most only those known to the inventor.

# How functional is your language?

- It depends on how closely you look at it.
- Known hardware is likely ok (propeller, stator, rotor, hanger, ADC).
- Does the language limit the way or result?
- Does your language exclude any other ways of solving the problem?
- Do you recite any essential steps?
- If it is functional, then you need structure in the specification

# If It Is M+F, What Is the Structure?

- Hardware cases are generally quite lenient on structure
  - Structure corresponding to "system memory means" is "a system memory" (*Chicago Board Option Exchange v. Int'l Securities Exchange*, 677 F.3d 1361 (Fed. Cir. 2012))
  - Structure corresponding to "arrangement for reactivating the link" is a processor and a transceiver even though neither was disclosed (*HTC Corp. v. IPCom GmbH*, 667 F.3d 1270 (Fed. Cir. 2012))
- But not always
  - "control device" was not sufficient structure to correspond to "programmable control means" element (*Ergo Licensing LLC v. Carefusion 303, Inc.*, 673 F.3d 1361, 1363-64 (Fed. Cir. 2012))

# What Is the Structure 2?

- In software, by contrast, we require an algorithm
  - *Aristocrat Techs. v. Int'l Game Tech.*, 521 F.3d 1328 (Fed. Cir. 2008)
  - *In re Aoyama*, 656 F.3d 1293 (Fed. Cir. 2011)
- If algorithm is present, claim is limited to that algorithm + equivalents
- If no algorithm, claim is indefinite under sec. 112(b)
- Even cases that are lenient on hardware require an algorithm. *HTC Corp. v. IPCom GmbH*, 667 F.3d 1270 (Fed. Cir. 2012) (raising it even though waived)

# What Is an Algorithm?

- In general, the Fed Cir wants specific computer programming
  - *Aristocrat Techs. v. Int'l Game Tech.*, 521 F.3d 1328 (Fed. Cir. 2008)
  - *In re Aoyama*, 656 F.3d 1293 (Fed. Cir. 2011)
  - *Ergo Licensing LLC v. Carefusion 303, Inc.*, 673 F.3d 1361, 1363-64 (Fed. Cir. 2012)
  - *ePlus, Inc. v. Lawson Software*, 700 F.3d 509 (Fed. Cir. Nov. 21, 2012) (specification must disclose structure, not just more function)
- But not always, especially when Judge Newman is on the panel
  - *In re Katz Interactive Call Processing Litig.*, 639 F.3d 1303 (Fed. Cir. 2011) (general purpose computer sufficient if no special programming required)
  - *Typhoon Touch Techs., Inc. v. Dell, Inc.,* 659 F.3d 1376 (Fed. Cir. 2011) (vague description of function sufficient)
- Algorithm that performs only some of the functions of the claim element insufficient
  - *Noah Sys. v. Intuit Inc.*, 675 F.3d 1302 (Fed. Cir. 2012) ("access means" not met by algorithm that discloses password system but not how to actually access the program once unlocked)

# Public Key Example 1

- Bob generates a public key and a private key
- Alice uses the public key to send a message to Bob
- Eve cannot read Alice’s message, even though she has the public key.
- Only Bob can read Alice’s message using his private key

# Public Key Example 2

- Can you claim sending secret messages without first meeting?
- Can you claim the outline?
- You need an algorithm to get the outline to work.
- Do you have to claim the knapsack algorithm?
- The claim is worth little if it does not cover the RSA algorithm.
- “computationally infeasible” to generate the message or the private key (Stanford Patent)
- Encrypting a message wherein breaking the code depends on the difficulty of factoring the product of large prime numbers
- Excludes quantum cryptography?