Future of Software IP: Unpatentability + Copyright

Carl Hewitt*

http://www.carlhewitt.info

@CarlHewitt on Twitter

Videos soon on YouTube, etc.!

©2013

*with the help of numerous colleagues
Outline

- Software copyright infringement criteria
- Software Patent Quagmire
- Proposed criteria of demarcation for software unpatentability
- Call for Action
Background on Copyright

Software IP

- Protection
  - Infringing similarity of expression
  - Many statutory & other exceptions

- Licensing
  - Automated clearing houses
    - Terms
    - Payment
Copyright Infringement Criteria

1. Abstraction
2. Filtration
3. Comparison
1. Abstraction

- Not restricted to literal elements of expression
  - Ontology
  - Structure
- Higher abstraction means less infringement
2. Filtration

Not infringing if derived from

- Efficiency considerations
- Scènes à faire
- External factors
  - APIs
  - Interoperability and compatibility
  - Design standards
  - Demands of market
  - Standard programming techniques
- Public domain
3. Comparison

Work as a whole

- Importance of infringing elements
- Economic damage of infringing elements
Software copyright protects **Expression** *not* **Process**

102(b) In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.
Software Patents

Quagmire?
Software Patent Extortion?

- “The Patent, Used as a Sword”
  NYT 10/7/2012
- “Are Patent Trolls Now Zeroed In On Start-Ups?”
  Forbes 1/17/2013
- “CentUp is Back Up on Indiegogo After Patent Troll Threat”
  EFF 2/25/2013
- Etc.
Patent for Patent Trolling!

IBM “System and method for extracting value from a portfolio of assets”*

Can IBM sell protection against being sued by a troll?

*US 8,386,350 issued 2/26/2013
Nature of Quagmire?

Is Operational Imprecision the primary problem?
Slippery Slope?

Mark Lemley has proposed that software patentability is OK if computational processes are precisely specified.*

Advanced Technology Programming Constructs

Highly Precise Operational Constructs

- Setting goals
- Strategies for achieving and assessing goals
- Conjectures, Metaphors & Analogies
- High-level executable contingency plans
- Argumentation
Advanced Programming Construct Examples

- Setting a goal
  ⊨ Differentiate[AIDS, Diabetes, differential]

- Strategy for achieving a goal
  when ⊨ Differentiate[diagnosis1, diagnosis2, differential] →
    ⊨ CanCause[morbidity, =diagnosis1],
    CanCause[morbidity, =diagnosis2] →
    ⊨ CommonCause[=morbidity, {=diagnosis1, =diagnosis2}] →
    ⊨ Differentiate[diagnosis1, diagnosis2, CommonMorbidity[morbidity]]

- Analogy
  ⊨ Analagous[SolarSystem ~ Atom,
              with {SolarSystem.sun ~ Atom.nucleus,
                  SolarSystem.planet ~ Atom.electron}]
Nature of Quagmire?

*Contra Lemley et al.
Nature of Quagmire:
Advanced Technology*
+ New Applications

*E.g. advanced programming constructs
"Bottomless Quagmire?"

- De Gaulle:
  "I predict you will sink step by step into a bottomless quagmire, however much you spend in men and money."

- US retort:
  French are incompetent; sour grapes!
Irresistible Pull of Software Patent Quagmire: Scalability for New Applications
Advanced Technology
Heuristic Knowledge

Millions of apps
- Each one potentially patentable
- Automated construction of patent applications?

- “There’s an app for that?”
Advanced Technology + New Applications

Consequence

Patenting heuristic knowledge of every intellectual field stands to overwhelm patent examiners & courts.
Overwhelmed!

- Patent office
  - Tsunami of patent applications
  - Will cost $37.50 for a micro-entity
- Software developers
  - Troll licensing
    - Own patents to trolls
    - External patents from trolls
  - Troll litigation + “Thermonuclear War”
- Courts
  - Troll litigation + “Thermonuclear War”
A way forward
Need Criteria for Demarcation

Fundamentally similar:

- Machine “thinking”
- Computational processes

“Bright red line”
Bright Red Line: Any of

Computation (machine “thinking”) comprised of

- Message sending
- Message receiving
- Message receiver creation
- Message processing*

*including how to process future messages
Why these criteria for *Bright Red Line*?

- **Video:** “Everything that you wanted to know about …” Hewitt, Meijer, and Szyperski. Microsoft Channel 9.

- **Scientific Papers:**
  - “*What is Computation?* …” Carl Hewitt. *in A Computable Universe*
    - [http://what-is-computation.carlhewitt.info](http://what-is-computation.carlhewitt.info)
Proposed Demarcation

Patentable claims

Abstract Computational Processes
(Machine “thinking”)
Process and Method

For each patent claim remove every element that consists of any of the following:

- Message sending
- Message receiving
- Message receiver creation
- Message processing*

*including how to process future messages
Patent Applied For

Application of Bright Red Line
Example:
A patent that crosses the line

IBM “Watson Jeopardy” patent claims:*  
• “Submitting a set of questions ...”  
  ➢ Message sending  
• “Receiving back ... a set of answers”  
  ➢ Message receiving  
• “Comparing the set of answers received ... to answers in ...”  
  ➢ Message processing

*US 2012/0077178 A1
Another Example:

Flook “Method for Updating Alarm Limits”:

- **Read process variables**
  - Message receiving: Physical signaling mechanism can be patented but not information (abstract meaning) conveyed

- **Calculate alarm limit**
  - Message processing: Physical substrate mechanism can be patented but not abstract computational process (machine “thinking”) induced

- **Set alarm bell**
  - Message sending: Physical signaling mechanism can be patented but not information (abstract meaning) conveyed

*US Ser. No. 194,032
Flook Analysis

Unpatentable

message receiving
message processing
message sending

"Bright red line"

Patentable

Sensors + Wires + Transceivers
Gates + Wires + Transceivers
Wires + Transceivers + Actuators
Possible Uses of Criteria

• USPTO to impose additional requirements in a manner similar to the current imposition of additional requirements for business method patents

• SCOTUS to demarcate software unpatentability
Partial Results

- **Gottschalk v. Benson**
  - A computational process to convert binary-coded decimal numbers into binary numbers
  - Ruled unpatentable

- **Parker v. Flook**
  - A computational process for updating an alarm limit (used to signal abnormal conditions) in a catalytic conversion process
  - Ruled unpatentable

Technology change mandates review!
Call to Action

Software UnPatentability Alliance

SUPA

Abolish software patentability

http://sup-alliance.org
SUPA

Potential alliance among:

- **Technology companies**
  - Large
    - "Thermonuclear War"
    - Swarms of independent inventions
  - Small
    - Financially crippling extortion
- **Investors**
  - Financial viability of start-ups?
- **Research labs**
- **Educational institutions**
- **Public interest groups**
- **Thought leaders**
“Software Patent Quagmire?” Panel Discussion

Eric Goldman, Santa Clara University (moderator)
Carl Hewitt, iRobust
Robert Merges, UC Berkeley
Tim Porter, Google
Pamela Samuelson, UC Berkeley

6PM April 24, 2013
http://www.vctaskforce.com/