The Trouble with Software Patents

Issues with the Patent System

- The patentability of software
- Patent quality
- Patents as a drag on innovation

Software Patents

- Should software be patentable?
- Is a program a “process, machine, manufacture, or composition of matter”?
- Does ordinary programming require far more day-to-day creativity than most other fields, and in turn affecting what one of “ordinary skill in the art” can do?
- Is there an adequate documentary record of prior art?
- Is 20 years far too long a period for such a dynamic field?
- Does the patent office have enough qualified people to evaluate software patents?
- But—why shouldn’t there be protection for something as unusual as, say, RSA?

Let’s Invent the First Thermostat, Circa 1950

- Find a physical phenomenon that changes with temperature: mercury expanding in a tube, a bimetallic bending because of different expansion rates, gallium and its alloys melting at a low temperature, conductivity changes with temperature, etc.
- Find a way to use this to control a furnace
- Patent it!

Let’s Invent the First Thermostat, Circa 2015

- Buy a solid state temperature sensor
- Add a microrocessor and a power control interface; program the microprocessor to monitor the temperature and control the furnace appropriately
- Patent it!

What is the Relationship Between These Patents?

- Is the second idea patentable if the first one already exists?
- (Let’s ignore the patent law issue of how the claims are written.)
- At the block diagram level, they’re the same
- Is “do it with a software” somehow different?

Computers Are Cheaper Today

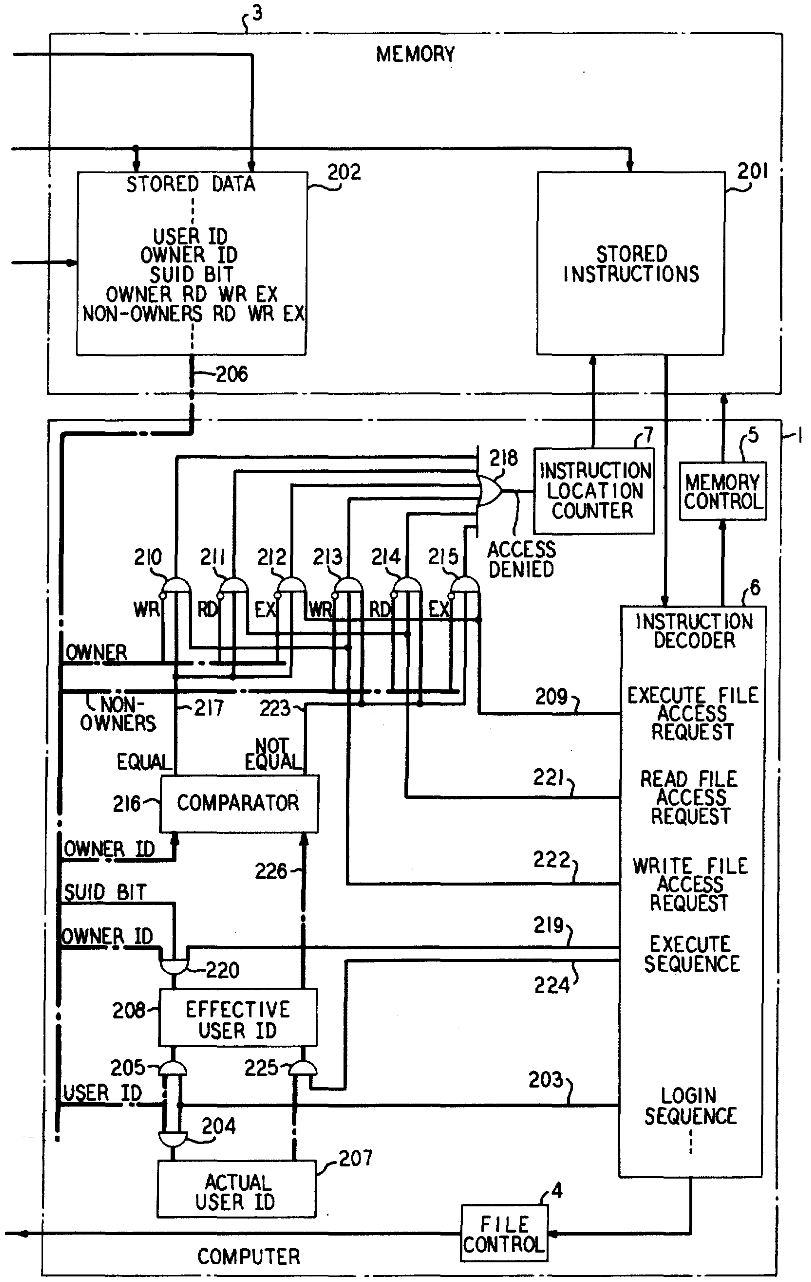
- Many things that were once done with circuitry are now done with microprocessors
- Example: integrators used to built with op amps, but today you can write some code instead
- It’s often much cheaper–but does the code somehow make it different?

Alice Corp. v. CLS Bank International

- Alice Corp's patents covered executing contracts through a computer, with the computer system acting as the escrow agent
- The Supreme Court: "Stating an abstract idea while adding the words 'apply it with a computer'" doesn't make something patentable
- That would seem to rule out our second thermostat patent
- But—most commenators found the Court's reasoning to be rather confused
- Does the Court understand technology well enough to actually rule on patents? Does Congress?

U.S. Patent Jan. 16, 1979 Sheet 2 of 2 4,135,240

FIG. 1B



Does the Patent Office Understand Programming?

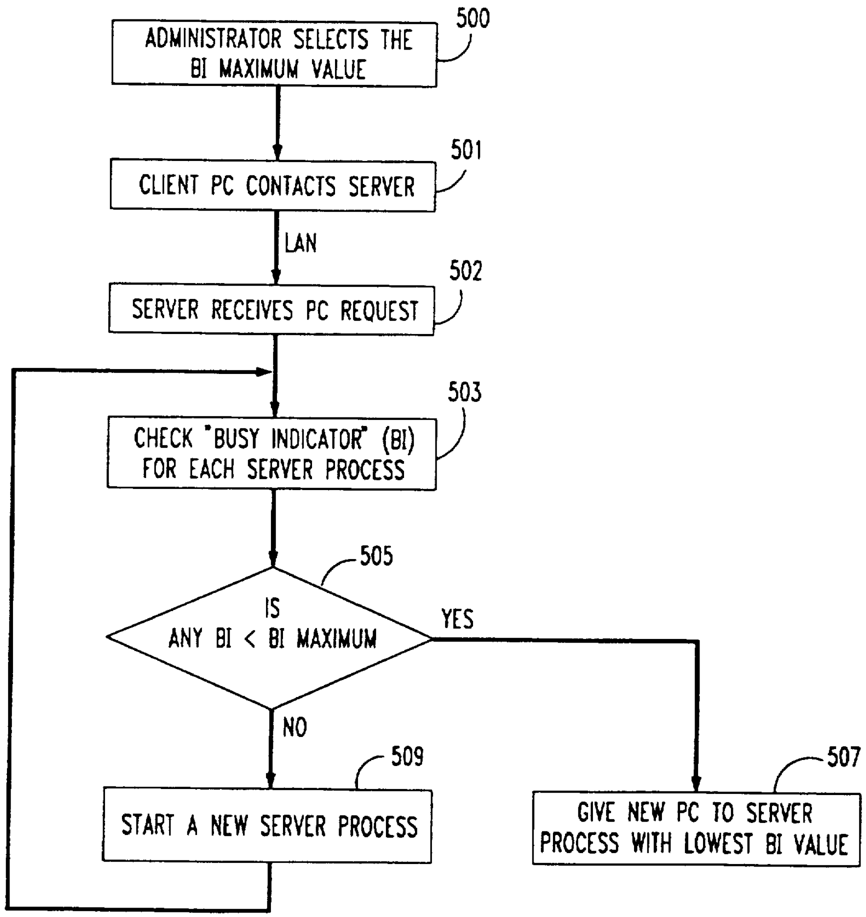
- They grant patents on things that any programmer can and would do
- They may not understand what stopped people from doing something earlier
- The result: bad patents

Method of and apparatus for operating a client/server computer network: US 5,249,290

A server apparatus for accessing one or more common resources using a plurality of server processes to which client service requests are assigned, said server apparatus comprising

> means for receiving an unassigned client service request requesting access to one of said common resources and
>
> means, responsive to a workload indication from each server process, each workload indication being less than a maximum workload for that server process, for assigning said unassigned received client service request to a server process having a workload indication which is less than the workload indication of all other server processes.

How It Works


500
ADMINISTRATOR SELECTS THE BI MAXIMUM VALUE
501
CLIENT PC CONTACTS SERVER
LAN
502
SERVER RECEIVES PC REQUEST
503
CHECK "BUSY INDICATOR" (BI) FOR EACH SERVER PROCESS
505
IS ANY BI < BI MAXIMUM
YES
NO
509
START A NEW SERVER PROCESS
507
GIVE NEW PC TO SERVER PROCESS WITH LOWEST BI VALUE

Why This Happens

- Not enough understanding of software by some examiners
- Not enough time spent on each patent
- Too little published prior art—most programmers don't bother filing patent applications on clever little tricks
- Also: the US patent office looks primarily at previous patents, and for many years software patents weren't even allowed—there's not enough history

But—There are Good Software Patents

- RSA—*still* the basis for most Internet encryption
- Non-obvious: it took Rivest, Shamir, and Adelman two years to devise their scheme, and Diffie and Hellman had tried and failed
- Easily showed as much creativity as most traditional patents
- Why shouldn't it be patentable?
- Note: the patent seriously interefered with use of crypto in Internet standards: the IETF didn't like patents

Patent “Trolls’

- Patent “trolls”—people who make money buying patents and suing corporations for infringement
- More formally known as “non-practicing entities” or “patent assertion entities”
- Issue: does the patent office do a good-enough job finding prior art or weeding out obvious ideas?
- Issue: effective priority date?
- Issue: should folks who have no interest in the invention per se be allowed to profit?

The Secondary Market

- Economists will tell you that the secondary market is important
- Are patent “trolls” any different than folks who buy stocks after the IPO?
- But—the constitutional purpose of patents is “[t]o promote the progress of science”. Do most inventors plan to sell the patent rights, as opposed to practicing the invention?
- That is, is the existence of the secondary market part of the actual incentive of the inventors?

Other Issues

- The problem is exacerbated by the number of bad software patents
- Patent “trolls” sometimes buy patents from bankrupt companies that had originally intended other uses for the patents
- There are some inventors and companies who develop ideas precisely to profit from licensing and/or lawsuits

Patent Trials

- Do judges understand the technology well enough to play their role?
- (Defining terms in the claims is a crucial part in patent lawsuits; the judge rules on that.)
- What about jurors' understanding?

How Companies Use Patents

- Directly, to exclude competitors from their market
- Bulk cross-licensing between two (typically large) companies
- Defensively: “we won’t assert our patents against you unless you sue us for patent infringement”
- Profit from licensing—or lawsuits

Settlements

- Patent lawsuits are extremely expensive, on both sides
- It may be cheaper to settle than to fight
- Defendants really win if they can prove “non-infringement” or “invalidity”
- They can also win in practice if the damages or licenses are very cheap

Standards-Essential Patents

- Often, industry standards rely on patented technology
- Example: https required use of RSA; until September 2000, it was patented
- Different standards organizations have different requirements:
  IEEE: “reasonable and non-discriminatory” (RAND) licensing
  W3C: “W3C will not approve a Recommendation if it is aware that Essential Claims exist which are not available on Royalty-Free terms.”
  IETF: participants must disclose patents’ existence

The Trouble with RAND

- What is “reasonable”?
- Is the royalty rate calculated on the covered feature, or on the overall price of the device? (Think about cellular-related patents used by smartphones.)
- What if a company reneges on its RAND commitment? This is generally actionable—but it can take a while

Non-Participants

- Standards bodies’ policies only bind participants
- Non-participants don’t have to share their patents, declare them openly, practice RAND, etc.
- Sometimes, the patent owner isn’t even aware of the standards proceeding at first

Do (Software) Patents Hurt Innovation?

- CS is very fast-moving—few ideas have that long a lifespan
- There are *many* patents; they're often dubious
- Are developers withdrawing from the market?
- Is it becoming impossible to innovate without risking a lawsuit? Remember that even winning a patent lawsuit is expensive.

Avoiding Patent Infringement

- It’s hard to know what is patented

☞ Searches aren’t easy

☞ Interpreting claims is very hard

☞ Knowing what portion of your idea or implementation may infringe some patent is almost impossible

- If you know of a patent and go ahead anyway, it may be “willful infringement”, and you may be liable for treble damages

☞ But you’re not required to search, so some companies bar their technical people from even looking

Why is Software Different?

- “Opportunistic licensers flourish when there is a large gap between the cost of getting a patent and the value that can be captured with an infringement action.” (Magliocca)
- Software is easy to create
- Patents aren’t that expensive, either
- The potential for profit is high