Are Software Patents Evil?

paulgraham.com · Paul Graham · 2006-03 · source

|
March 2006

(This essay is derived from a talk at Google.)

A few weeks ago I found to my surprise that I'd been granted four patents. This was all the more surprising because I'd only applied for three. The patents aren't mine, of course. They were assigned to Viaweb, and became Yahoo's when they bought us. But the news set me thinking about the question of software patents generally.

Patents are a hard problem. I've had to advise most of the startups we've funded about them, and despite years of experience I'm still not always sure I'm giving the right advice.

One thing I do feel pretty certain of is that if you're against software patents, you're against patents in general. Gradually our machines consist more and more of software. Things that used to be done with levers and cams and gears are now done with loops and trees and closures. There's nothing special about physical embodiments of control systems that should make them patentable, and the software equivalent not.

Unfortunately, patent law is inconsistent on this point. Patent law in most countries says that algorithms aren't patentable. This rule is left over from a time when "algorithm" meant something like the Sieve of Eratosthenes. In 1800, people could not see as readily as we can that a great many patents on mechanical objects were really patents on the algorithms they embodied.

Patent lawyers still have to pretend that's what they're doing when they patent algorithms. You must not use the word "algorithm" in the title of a patent application, just as you must not use the

word "essays" in the title of a book. If you want to patent an algorithm, you have to frame it as a computer system executing that algorithm. Then it's mechanical; phew. The default euphemism for algorithm is "system and method." Try a patent search for that phrase and see how many results you get.

Since software patents are no different from hardware patents, people who say "software patents are evil" are saying simply "patents are evil." So why do so many people complain about software patents specifically?

I think the problem is more with the patent office than the concept of software patents. Whenever software meets government, bad things happen, because software changes fast and government changes slow. The patent office has been overwhelmed by both the volume and the novelty of applications for software patents, and as a result they've made a lot of mistakes.

The most common is to grant patents that shouldn't be granted. To be patentable, an invention has to be more than new. It also has to be non-obvious. And this, especially, is where the USPTO has been dropping the ball. Slashdot has an icon that expresses the problem vividly: a knife and fork with the words "patent pending" superimposed.

The scary thing is, this is the only icon they have for patent stories. Slashdot readers now take it for granted that a story about a patent will be about a bogus patent. That's how bad the problem has become.

The problem with Amazon's notorious one-click patent, for example, is not that it's a software patent, but that it's obvious. Any online store that kept people's shipping addresses would have implemented this. The reason Amazon did it first was not that they were especially smart, but because they were one of the earliest sites with enough clout to force customers to log in before they could buy something.
[1]

We, as hackers, know the USPTO is letting people patent the knives

and forks of our world. The problem is, the USPTO are not hackers. They're probably good at judging new inventions for casting steel or grinding lenses, but they don't understand software yet.

At this point an optimist would be tempted to add "but they will eventually." Unfortunately that might not be true. The problem with software patents is an instance of a more general one: the patent office takes a while to understand new technology. If so, this problem will only get worse, because the rate of technological change seems to be increasing. In thirty years, the patent office may understand the sort of things we now patent as software, but there will be other new types of inventions they understand even less.

Applying for a patent is a negotiation. You generally apply for a broader patent than you think you'll be granted, and the examiners reply by throwing out some of your claims and granting others. So I don't really blame Amazon for applying for the one-click patent. The big mistake was the patent office's, for not insisting on something narrower, with real technical content. By granting such an over-broad patent, the USPTO in effect slept with Amazon on the first date. Was Amazon supposed to say no?

Where Amazon went over to the dark side was not in applying for the patent, but in enforcing it. A lot of companies (Microsoft, for example) have been granted large numbers of preposterously over-broad patents, but they keep them mainly for defensive purposes. Like nuclear weapons, the main role of big companies' patent portfolios is to threaten anyone who attacks them with a counter-suit. Amazon's suit against Barnes & Noble was thus the equivalent of a nuclear first strike.

That suit probably hurt Amazon more than it helped them. Barnes & Noble was a lame site; Amazon would have crushed them anyway. To attack a rival they could have ignored, Amazon put a lasting black mark on their own reputation. Even now I think if you asked hackers to free-associate about Amazon, the one-click patent would turn up in the first ten topics.

Google clearly doesn't feel that merely holding patents is evil.

They've applied for a lot of them. Are they hypocrites? Are patents evil?

There are really two variants of that question, and people answering it often aren't clear in their own minds which they're answering. There's a narrow variant: is it bad, given the current legal system, to apply for patents? and also a broader one: is it bad that the current legal system allows patents?

These are separate questions. For example, in preindustrial societies like medieval Europe, when someone attacked you, you didn't call the police. There were no police. When attacked, you were supposed to fight back, and there were conventions about how to do it. Was this wrong? That's two questions: was it wrong to take justice into your own hands, and was it wrong that you had to? We tend to say yes to the second, but no to the first. If no one else will defend you, you have to defend yourself.
[2]

The situation with patents is similar. Business is a kind of ritualized warfare. Indeed, it evolved from actual warfare: most early traders switched on the fly from merchants to pirates depending on how strong you seemed. In business there are certain rules describing how companies may and may not compete with one another, and someone deciding that they're going to play by their own rules is missing the point. Saying "I'm not going to apply for patents just because everyone else does" is not like saying "I'm not going to lie just because everyone else does." It's more like saying "I'm not going to use TCP/IP just because everyone else does." Oh yes you are.

A closer comparison might be someone seeing a hockey game for the first time, realizing with shock that the players were deliberately bumping into one another, and deciding that one would on no account be so rude when playing hockey oneself.

Hockey allows checking. It's part of the game. If your team refuses to do it, you simply lose. So it is in business. Under the present rules, patents are part of the game.

What does that mean in practice? We tell the startups we fund not to worry about infringing patents, because startups rarely get sued for patent infringement. There are only two reasons someone might sue you: for money, or to prevent you from competing with them. Startups are too poor to be worth suing for money. And in practice they don't seem to get sued much by competitors, either. They don't get sued by other startups because (a) patent suits are an expensive distraction, and (b) since the other startups are as young as they are, their patents probably haven't issued yet.
[3]
Nor do startups,
at least in the software business, seem to get sued much by established competitors. Despite all the patents Microsoft holds, I don't know of an instance where they sued a startup for patent infringement. Companies like Microsoft and Oracle don't win by winning lawsuits. That's too uncertain. They win by locking competitors out of their sales channels. If you do manage to threaten them, they're more likely to buy you than sue you.

When you read of big companies filing patent suits against smaller ones, it's usually a big company on the way down, grasping at straws. For example, Unisys's attempts to enforce their patent on LZW compression. When you see a big company threatening patent suits, sell. When a company starts fighting over IP, it's a sign they've lost the real battle, for users.

A company that sues competitors for patent infringement is like a defender who has been beaten so thoroughly that he turns to plead with the referee. You don't do that if you can still reach the ball, even if you genuinely believe you've been fouled. So a company threatening patent suits is a company in trouble.

When we were working on Viaweb, a bigger company in the e-commerce business was granted a patent on online ordering, or something like that. I got a call from a VP there asking if we'd like to license it. I replied that I thought the patent was completely bogus, and would never hold up in court. "Ok," he replied. "So, are you guys hiring?"

If your startup grows big enough, however, you'll start to get sued,

no matter what you do. If you go public, for example, you'll be sued by multiple patent trolls who hope you'll pay them off to go away. More on them later.

In other words, no one will sue you for patent infringement till you have money, and once you have money, people will sue you whether they have grounds to or not. So I advise fatalism. Don't waste your time worrying about patent infringement. You're probably violating a patent every time you tie your shoelaces. At the start, at least, just worry about making something great and getting lots of users. If you grow to the point where anyone considers you worth attacking, you're doing well.

We do advise the companies we fund to apply for patents, but not so they can sue competitors. Successful startups either get bought or grow into big companies. If a startup wants to grow into a big company, they should apply for patents to build up the patent portfolio they'll need to maintain an armed truce with other big companies. If they want to get bought, they should apply for patents because patents are part of the mating dance with acquirers.

Most startups that succeed do it by getting bought, and most acquirers care about patents. Startup acquisitions are usually a build-vs-buy decision for the acquirer. Should we buy this little startup or build our own? And two things, especially, make them decide not to build their own: if you already have a large and rapidly growing user base, and if you have a fairly solid patent application on critical parts of your software.

There's a third reason big companies should prefer buying to building: that if they built their own, they'd screw it up. But few big companies are smart enough yet to admit this to themselves. It's usually the acquirer's engineers who are asked how hard it would be for the company to build their own, and they overestimate their abilities.
[4]
A patent seems to change the balance. It gives the acquirer an excuse to admit they couldn't copy what you're doing. It may also help them to grasp what's special about your technology.

Frankly, it surprises me how small a role patents play in the software business. It's kind of ironic, considering all the dire things experts say about software patents stifling innovation, but when one looks closely at the software business, the most striking thing is how little patents seem to matter.

In other fields, companies regularly sue competitors for patent infringement. For example, the airport baggage scanning business was for many years a cozy duopoly shared between two companies, InVision and L-3. In 2002 a startup called Reveal appeared, with new technology that let them build scanners a third the size. They were sued for patent infringement before they'd even released a product.

You rarely hear that kind of story in our world. The one example I've found is, embarrassingly enough, Yahoo, which filed a patent suit against a gaming startup called Xfire in 2005. Xfire doesn't seem to be a very big deal, and it's hard to say why Yahoo felt threatened. Xfire's VP of engineering had worked at Yahoo on similar stuff-- in fact, he was listed as an inventor on the patent Yahoo sued over-- so perhaps there was something personal about it. My guess is that someone at Yahoo goofed. At any rate they didn't pursue the suit very vigorously.

Why do patents play so small a role in software? I can think of three possible reasons.

One is that software is so complicated that patents by themselves are not worth very much. I may be maligning other fields here, but it seems that in most types of engineering you can hand the details of some new technique to a group of medium-high quality people and get the desired result. For example, if someone develops a new process for smelting ore that gets a better yield, and you assemble a team of qualified experts and tell them about it, they'll be able to get the same yield. This doesn't seem to work in software. Software is so subtle and unpredictable that "qualified experts" don't get you very far.

That's why we rarely hear phrases like "qualified expert" in the software business. What that level of ability can get you is, say,

to make your software compatible with some other piece of software--in eight months, at enormous cost. To do anything harder you need individual brilliance. If you assemble a team of qualified experts and tell them to make a new web-based email program, they'll get their asses kicked by a team of inspired nineteen year olds.

Experts can implement, but they can't design.
Or rather, expertise in implementation is the only kind most people, including the experts themselves, can measure.
[5]

But design is a definite skill. It's not just an airy intangible. Things always seem intangible when you don't understand them. Electricity seemed an airy intangible to most people in 1800. Who knew there was so much to know about it? So it is with design. Some people are good at it and some people are bad at it, and there's something very tangible they're good or bad at.

The reason design counts so much in software is probably that there are fewer constraints than on physical things. Building physical things is expensive and dangerous. The space of possible choices is smaller; you tend to have to work as part of a larger group; and you're subject to a lot of regulations. You don't have any of that if you and a couple friends decide to create a new web-based application.

Because there's so much scope for design in software, a successful application tends to be way more than the sum of its patents. What protects little companies from being copied by bigger competitors is not just their patents, but the thousand little things the big company will get wrong if they try.

The second reason patents don't count for much in our world is that startups rarely attack big companies head-on, the way Reveal did. In the software business, startups beat established companies by transcending them. Startups don't build desktop word processing programs to compete with Microsoft Word.
[6]
They build Writely.
If this paradigm is crowded, just wait for the next one; they run

pretty frequently on this route.

Fortunately for startups, big companies are extremely good at denial. If you take the trouble to attack them from an oblique angle, they'll meet you half-way and maneuver to keep you in their blind spot. To sue a startup would mean admitting it was dangerous, and that often means seeing something the big company doesn't want to see. IBM used to sue its mainframe competitors regularly, but they didn't bother much about the microcomputer industry because they didn't want to see the threat it posed. Companies building web based apps are similarly protected from Microsoft, which even now doesn't want to imagine a world in which Windows is irrelevant.

The third reason patents don't seem to matter very much in software is public opinion-- or rather, hacker opinion. In a recent interview, Steve Ballmer coyly left open the possibility of attacking Linux on patent grounds. But I doubt Microsoft would ever be so stupid. They'd face the mother of all boycotts. And not just from the technical community in general; a lot of their own people would rebel.

Good hackers care a lot about matters of principle, and they are highly mobile. If a company starts misbehaving, smart people won't work there. For some reason this seems to be more true in software than other businesses. I don't think it's because hackers have intrinsically higher principles so much as that their skills are easily transferrable. Perhaps we can split the difference and say that mobility gives hackers the luxury of being principled.

Google's "don't be evil" policy may for this reason be the most valuable thing they've discovered. It's very constraining in some ways. If Google does do something evil, they get doubly whacked for it: once for whatever they did, and again for hypocrisy. But I think it's worth it. It helps them to hire the best people, and it's better, even from a purely selfish point of view, to be constrained by principles than by stupidity.

(I wish someone would get this point across to the present administration.)

I'm not sure what the proportions are of the preceding three ingredients, but the custom among the big companies seems to be not to sue the small ones, and the startups are mostly too busy and too poor to sue one another. So despite the huge number of software patents there's not a lot of suing going on. With one exception: patent trolls.

Patent trolls are companies consisting mainly of lawyers whose whole business is to accumulate patents and threaten to sue companies who actually make things. Patent trolls, it seems safe to say, are evil. I feel a bit stupid saying that, because when you're saying something that Richard Stallman and Bill Gates would both agree with, you must be perilously close to tautologies.

The CEO of Forgent, one of the most notorious patent trolls, says that what his company does is "the American way." Actually that's not true. The American way is to make money by creating wealth, not by suing people.
[7]
What companies like Forgent do is actually the proto-industrial way. In the period just before the industrial revolution, some of the greatest fortunes in countries like England and France were made by courtiers who extracted some lucrative right from the crown-- like the right to collect taxes on the import of silk-- and then used this to squeeze money from the merchants in that business. So when people compare patent trolls to the mafia, they're more right than they know, because the mafia too are not merely bad, but bad specifically in the sense of being an obsolete business model.

Patent trolls seem to have caught big companies by surprise. In the last couple years they've extracted hundreds of millions of dollars from them. Patent trolls are hard to fight precisely because they create nothing. Big companies are safe from being sued by other big companies because they can threaten a counter-suit. But because patent trolls don't make anything, there's nothing they can be sued for. I predict this loophole will get closed fairly quickly, at least by legal standards. It's clearly an abuse of the system, and the victims are powerful.
[8]

But evil as patent trolls are, I don't think they hamper innovation

much. They don't sue till a startup has made money, and by that point the innovation that generated it has already happened. I can't think of a startup that avoided working on some problem because of patent trolls.

So much for hockey as the game is played now. What about the more theoretical question of whether hockey would be a better game without checking? Do patents encourage or discourage innovation?

This is a very hard question to answer in the general case. People write whole books on the topic. One of my main hobbies is the history of technology, and even though I've studied the subject for years, it would take me several weeks of research to be able to say whether patents have in general been a net win.

One thing I can say is that 99.9% of the people who express opinions on the subject do it not based on such research, but out of a kind of religious conviction. At least, that's the polite way of putting it; the colloquial version involves speech coming out of organs not designed for that purpose.

Whether they encourage innovation or not, patents were at least intended to. You don't get a patent for nothing. In return for the exclusive right to use an idea, you have to publish it, and it was largely to encourage such openness that patents were established.

Before patents, people protected ideas by keeping them secret. With patents, central governments said, in effect, if you tell everyone your idea, we'll protect it for you. There is a parallel here to the rise of civil order, which happened at roughly the same time. Before central governments were powerful enough to enforce order, rich people had private armies. As governments got more powerful, they gradually compelled magnates to cede most responsibility for protecting them. (Magnates still have bodyguards, but no longer to protect them from other magnates.)

Patents, like police, are involved in many abuses. But in both cases the default is something worse. The choice is not "patents or freedom?" any more than it is "police or freedom?" The actual

questions are respectively "patents or secrecy?" and "police or gangs?"

As with gangs, we have some idea what secrecy would be like, because that's how things used to be. The economy of medieval Europe was divided up into little tribes, each jealously guarding their privileges and secrets. In Shakespeare's time, "mystery" was synonymous with "craft." Even today we can see an echo of the secrecy of medieval guilds, in the now pointless secrecy of the Masons.

The most memorable example of medieval industrial secrecy is probably Venice, which forbade glassblowers to leave the city, and sent assassins after those who tried. We might like to think we wouldn't go so far, but the movie industry has already tried to pass laws prescribing three year prison terms just for putting movies on public networks. Want to try a frightening thought experiment? If the movie industry could have any law they wanted, where would they stop? Short of the death penalty, one assumes, but how close would they get?

Even worse than the spectacular abuses might be the overall decrease in efficiency that would accompany increased secrecy. As anyone who has dealt with organizations that operate on a "need to know" basis can attest, dividing information up into little cells is terribly inefficient. The flaw in the "need to know" principle is that you don't know who needs to know something. An idea from one area might spark a great discovery in another. But the discoverer doesn't know he needs to know it.

If secrecy were the only protection for ideas, companies wouldn't just have to be secretive with other companies; they'd have to be secretive internally. This would encourage what is already the worst trait of big companies.

I'm not saying secrecy would be worse than patents, just that we couldn't discard patents for free. Businesses would become more secretive to compensate, and in some fields this might get ugly. Nor am I defending the current patent system. There is clearly a lot that's broken about it. But the breakage seems to affect

software less than most other fields.

In the software business I know from experience whether patents encourage or discourage innovation, and the answer is the type that people who like to argue about public policy least like to hear: they don't affect innovation much, one way or the other. Most innovation in the software business happens in startups, and startups should simply ignore other companies' patents. At least, that's what we advise, and we bet money on that advice.

The only real role of patents, for most startups, is as an element of the mating dance with acquirers. There patents do help a little. And so they do encourage innovation indirectly, in that they give more power to startups, which is where, pound for pound, the most innovation happens. But even in the mating dance, patents are of secondary importance. It matters more to make something great and get a lot of users.

Notes

[1]
You have to be careful here, because a great discovery often seems obvious in retrospect. One-click ordering, however, is not such a discovery.

[2]
"Turn the other cheek" skirts the issue; the critical question is not how to deal with slaps, but sword thrusts.

[3]
Applying for a patent is now very slow, but it might actually be bad if that got fixed. At the moment the time it takes to get a patent is conveniently just longer than the time it takes a startup to succeed or fail.

[4]
Instead of the canonical "could you build this?" maybe the corp dev guys should be asking "will you build this?" or even "why haven't

you already built this?"

[5]
Design ability is so hard to measure that you can't even trust the design world's internal standards. You can't assume that someone with a degree in design is any good at design, or that an eminent designer is any better than his peers. If that worked, any company could build products as good as Apple's just by hiring sufficiently qualified designers.

[6]
If anyone wanted to try, we'd be interested to hear from them. I suspect it's one of those things that's not as hard as everyone assumes.

[7]
Patent trolls can't even claim, like speculators, that they "create" liquidity.

[8]
If big companies don't want to wait for the government to take action, there is a way to fight back themselves. For a long time I thought there wasn't, because there was nothing to grab onto. But there is one resource patent trolls need: lawyers. Big technology companies between them generate a lot of legal business. If they agreed among themselves never to do business with any firm employing anyone who had worked for a patent troll, either as an employee or as outside counsel, they could probably starve the trolls of the lawyers they need.

Thanks to Dan Bloomberg, Paul Buchheit, Sarah Harlin, Jessica Livingston, and Peter Norvig for reading drafts of this, to Joel Lehrer and Peter Eng for answering my questions about patents, and to Ankur Pansari for inviting me to speak.

|