



## *Opinion: The problem with software patents? They don't scale*

By Timothy B. Lee and Christina Mulligan

March 8, 2012

*Timothy B. Lee, a regular contributor to Ars Technica and an adjunct scholar at the Cato Institute, has joined forces with Christina Mulligan, a postdoctoral resident fellow at the Information Society Project at Yale Law School, to research a key problem posed by software patents: the cost of finding out that they exist. This op-ed distills their most recent research paper. The views expressed here do not necessarily represent those of Ars Technica.*

Nathan Myhrvold, the Microsoft veteran who founded the patent-trolling giant Intellectual Ventures, [loves to complain](#) about the "culture of intentionally infringing patents" in the software industry. "You have a set of people who are used to getting something for free," he told *Business Week* in 2006.

Myhrvold is right that patent infringement is rampant among software firms. But in demanding that this infringement stop, Myhrvold isn't just declaring war on what he regards as Silicon Valley's patent-hostile culture. He's declaring war on the laws of mathematics. The legal research required for all software-producing firms to stop infringing patents would cost more than the entire revenue of the software industry. Even if firms were willing to pay the bill, there simply aren't enough patent lawyers to do the work. Firms infringe software patents because they don't have any other choice.

If a real estate developer wants to build on a particular piece of land, she first must figure out who owns the land before she can negotiate a contract and start construction. Most of the time, this is easy. The landowner can be readily identified in a public records office.

In principle, a software developer starting a new project faces a similar problem. He needs to know if the software he is planning to create will accidentally infringe on anyone's patents. But whereas looking up who holds claims to a particular piece of land is easy, finding out who, if anyone, holds patents related to a particular piece of software is difficult and expensive. It's so difficult, in fact, that the vast majority of software developers [don't even try](#).

Why is software different from real estate? In a [new paper](#), we argue the fundamental difference is a matter of scalability: how much effort it takes to discover who owns an invention—or a piece of land—as the number of patents or land parcels increases. Property rights in land scale well because parcels exist in relatively well-defined locations on a two-dimensional plane. County officials take advantage of this fact to store records in a predictable order (or, more recently, to build databases searchable by geographical location). Geographical locations serve as an "index" for real property claims, so record-keepers can find any specific file quickly no matter how many files there are.

Some patents are similarly "indexable." Chemical patents, for example, can be organized by chemical formula. Indeed, a German organization called FIZ Karlsruhe offers an [electronic database called STN](#) which allows researchers to look up patents based on their chemical formula. The existence of products like STN is one reason patent litigation is [much less common](#) for chemical patents than for software patents.

Unfortunately, software patents don't scale well. It doesn't seem possible to create an STN-like database for software patents. There's nothing analogous to geographical coordinates or a chemical formula to uniquely identify software inventions. It's hard to predict which aspects of a software product someone might try to patent. It's even harder to predict which terms a patent lawyer might use to describe these concepts. So searching by keywords is likely to uncover only a fraction of relevant patents.

This means the only foolproof way to find all the patents that cover a particular computer program is by "brute force"—to pay a patent lawyer to sift through every software patent, one at a time, looking for ones that might be relevant. There are hundreds of thousands of software patents, with 40,000 new ones released every year. A single firm could easily spend hundreds of thousands of dollars on patent research for a single software product.

## **Everyone's problem**

Who needs to worry about infringing software patents? The Green Bay Packers, OfficeMax, Kraft Foods, Aeropostale, and Oprah Winfrey's Harpo productions have all faced software patent lawsuits. Indeed, virtually every medium and large firm in the United States performs activities—like maintaining a public website, using a

computerized point-of-sale system, or using an Internet-based invoicing system—likely to infringe some software patents. So all of these firms are part of the software industry, at least as far as patent law is concerned.

In our paper, we estimate it would take at least 2,000,000 patent attorneys, working full time, to consider whether all these software-producing firms have infringed any of the software patents issued in a typical year. Even if firms wanted to hire that many attorneys, they couldn't; there are only 40,000 registered patent attorneys and agents in the United States.

So Myhrvold is wrong to suggest firms should ignore software patents because they're trying to "get something for free." They ignore software patents because it's mathematically impossible for them to do anything else. The patent system simply doesn't scale up to an industry as complex and decentralized as the software industry. And if it's practically impossible for firms to avoid infringing software patents, it's unfair to punish them for their failure.