Anatomy of a Trivial Patent

GNU philosophy

This essay was originally published on http://gnu.org, in 2006.

This document is part of GNU philosophy, the GNU Project's exhaustive collection of articles and essays about free software and related matters.

Anatomy of a Trivial Patent

Programmers are well aware that many of the existing software patents cover laughably obvious ideas. Yet the patent system's defenders often argue that these ideas are nontrivial, obvious only in hindsight. And it is surprisingly difficult to defeat them in debate. Why is that?

One reason is that any idea can be made to look complex when analyzed to death. Another reason is that these trivial ideas often look quite complex as described in the patents themselves. The patent system's defenders can point to the complex description and say, "How can anything this complex be obvious?"

I will use an example to show you how. Here's claim number one from US patent number 5,963,916, applied for in October 1996:

> 1. A method for enabling a remote user to preview a portion of a pre-recorded music product from a network web site containing pre-selected portions of different pre-recorded music products, using a computer, a computer display and a telecommunications link between the remote user's computer and the network web site, the method comprising the steps of:
>
> - using the remote user's computer to establish a telecommunications link to the network web site wherein the network web site comprises (i) a central host server coupled to a communications network for retrieving and transmitting the pre-selected portion of the pre-recorded music product upon request by a remote user and (ii) a central storage device for storing pre-selected portions of a plurality of different pre-recorded music products;
> - transmitting user identification data from the remote user's computer to the central host server thereby allowing the central host server to identify and track the user's progress through the network web site;
> - choosing at least one pre-selected portion of the pre-recorded music products from the central host server;
> - receiving the chosen pre-selected portion of the pre-recorded products; and
> - interactively previewing the received chosen pre-selected portion of the pre-recorded music product.

That sure looks like a complex system, right? Surely it took a real clever guy to think of this? No, but it took cleverness to make it seem so complex. Let's analyze where the complexity comes from:

> 1. A method for enabling a remote user to preview a portion of a pre-recorded music product from a network web site containing pre-selected portions

That states the principal part of their idea. They put selections from certain pieces of music on a server so a user can listen to them.

> of different pre-recorded music products,

This emphasizes their server stores selections from more than one piece of music.

It is a basic principle of computer science is that if a computer can do a thing once, it can do that thing many times, on different data each time. Many patents pretend that applying this principle to a specific case makes an "invention."

> using a computer, a computer display and a telecommunications link between the remote user's computer and the network web site,

This says they are using a server on a network.

> the method comprising the steps of:

- using the remote user's computer to establish a telecommunications link to the network web site

This says that the user connects to the server over the network. (That's the way one uses a server.)

> wherein the network web site comprises (i) a central host server coupled to a communications network

This informs us that the server is on the net. (That is typical of servers.)

> for retrieving and transmitting the pre-selected portion of the pre-recorded music product upon request by a remote user

This repeats the general idea stated in the first two lines.

> and (ii) a central storage device for storing pre-selected portions of a plurality of different pre-recorded music products;

They have decided to put a hard disk (or equivalent) in their computer and store the music samples on that. Ever since around 1980, this has been the normal way to store anything on a computer for rapid access.

Note how they emphasize once again the fact that they can store more than one selection on this disk. Of course, every file system will let you store more than one file.

- transmitting user identification data from the remote user's computer to the central host server thereby allowing the central host server to identify and track the user's progress through the network web site;

This says that they keep track of who you are and what you access—a common (though nasty) thing for web servers to do. I believe it was common already in 1996.

- choosing at least one pre-selected portion of the pre-recorded music products from the central host server;

In other words, the user clicks to say which link to follow. That is typical for web servers; if they had found another way to do it, that might have been an invention.

- receiving the chosen pre-selected portion of the pre-recorded products; and

When you follow a link, your browser reads the contents. This is typical behavior for a web browser.

- interactively previewing the received chosen pre-selected portion of the pre-recorded music product.

This says that your browser plays the music for you. (That is what many browsers do, when you follow a link to an audio file.)

Now you see how they padded this claim to make it into a complex idea: they combined their own idea (stated in two lines of text) with important aspects of what computers, networks, web servers, and web browsers do. This adds up to the so-called invention for which they received the patent.

This example is typical of software patents. Even the occasional patent whose idea is nontrivial has the same sort of added complication.

Now look at a subsequent claim:

> 3. The method of [149]claim 1 wherein the central memory device comprises a plurality of compact disc-read only memory (CD-ROMs).

What they are saying here is, "Even if you don't think that claim 1 is really an invention, using CD-ROMs to store the data makes it an invention for sure. An average system designer would never have thought of storing data on a CD."

Now look at the next claim:

> 4. The method of [150]claim 1 wherein the central memory device comprises a RAID array drive.

A RAID array is a group of disks set up to work like one big disk, with the special feature that, even if one of the disks in the array has a failure and stops working, all the data are still available on the other disks in the group. Such arrays have been commercially available since long before 1996, and are a standard way of storing data for high availability. But these brilliant inventors have patented the use of a RAID array for this particular purpose.

Trivial as it is, this patent would not necessarily be found legally invalid if there is a lawsuit about it. Not only the US Patent Office but the courts as well tend to apply a very low standard when judging whether a patent is "unobvious." This patent might pass muster, according to them.

What's more, the courts are reluctant to overrule the Patent Office, so there is a better chance of getting a patent overturned if you can show a court prior art that the Patent Office did not consider. If the courts are willing to entertain a higher standard in judging unobviousness, it helps to save the prior art for them. Thus, the proposals to "make the system work better" by providing the Patent Office with a better database of prior art could instead make things worse.

It is very hard to make a patent system behave reasonably; it is a complex bureaucracy and tends to follow its structural imperatives regardless of what it is "supposed" to do. The only practical way to get rid of the many obvious patents on software features and business practices is to get rid of all patents in those fields. Fortunately, that would be no loss: the unobvious patents in the software field do no good either. What software patents do is put software developers and users under threat.

The patent system is supposed, intended, to promote progress, and those who benefit from software patents ask us to believe without question that they do have that effect. But programmers' experience shows otherwise. New theoretical analysis shows that this is no paradox. (See `http://researchoninnovation.org/patent.pdf`.) There is no reason why society should expose software developers and users to the danger of software patents.