

Patent absurdity¹

If patent law had been applied to novels in the 1880s, great books would not have been written. If the EU applies it to software, every computer user will be restricted, says Richard Stallman

Next month, the European Parliament will vote on the vital question of whether to allow patents covering software, which would restrict every computer user and tie software developers up in knots.

Many politicians may be voting blindly - not being programmers, they don't understand what software patents do. They often think patents are similar to copyright law (except for some details), which is not the case.

For example, when I publicly asked Patrick Devedjian, then the minister for industry, how France would vote on the issue of software patents, he responded with an impassioned defence of copyright law, praising Victor Hugo for his role in the adoption of copyright.

Those who imagine effects like those of copyright law cannot grasp the real effects of software patents. We can use Hugo as an example to illustrate the difference between the two.

A novel and a modern complex programme have certain points in common: each is large and implements many ideas. Suppose patent law had been applied to novels in the 1800s; suppose states such as France had permitted the patenting of literary ideas. How would this have affected Hugo's writing? How would the effects of literary patents compare with the effects of literary copyright?

Consider the novel *Les Misérables*, written by Hugo. Because he wrote it, the copyright belonged only to him. He did not have to fear that some stranger could sue him for copyright infringement and win. That was impossible, because copyright covers only the details of a work of authorship, and only restricts copying. Hugo had not copied *Les Misérables*, so he was not in danger.

Patents work differently. They cover ideas - each patent is a monopoly on practising some idea, which is described in the patent itself.

Here's one example of a hypothetical literary patent:

Claim 1: a communication process that represents, in the mind of a reader, the concept of a character who has been in jail for a long time and becomes bitter towards society and humankind.

Claim 2: a communication process according to claim 1, wherein said character subsequently finds moral redemption through the kindness of another.

¹ Author: Richard Stallman, printed in *The Guardian*, 6/25/05.

Claim 3: a communication process according to claims 1 and 2, wherein said character changes his name during the story.

If such a patent had existed in 1862 when *Les Misérables* was published, the novel would have infringed all three claims - all these things happened to Jean Valjean in the novel. Hugo could have been sued, and would have lost. The novel could have been prohibited - in effect, censored - by the patent holder.

Now consider this hypothetical literary patent:

Claim 1: a communication process that represents, in the mind of a reader, the concept of a character who has been in jail for a long time and subsequently changes his name.

Les Misérables would have infringed that patent too, because it also fits the life story of Jean Valjean.

These patents would all cover the story of one character in a novel. They overlap, but they do not precisely duplicate each other, so they could all be valid simultaneously - all the patent holders could have sued Victor Hugo. Any one of them could have prohibited publication of *Les Misérables*.

You might think these ideas are so simple that no patent office would have issued them. We programmers are often amazed by the simplicity of the ideas that real software patents cover - for instance, the European Patent Office has issued a patent on the progress bar, and one on accepting payment via credit cards. These would be laughable if they were not so dangerous.

Other aspects of *Les Misérables* could also have fallen foul of patents. For instance, there could have been a patent on a fictionalised portrayal of the Battle of Waterloo, or a patent on using Parisian slang in fiction. Two more lawsuits.

In fact, there is no limit to the number of different patents that might have been applicable for suing the author of a work like *Les Misérables*. All the patent holders would claim they deserved a reward for the literary progress that their patented ideas represented - but these obstacles would not promote progress in literature. They would only obstruct it.

However, a very broad patent could have made all these issues irrelevant. Imagine patents with broad claims, like these:

Communication process structured with narration that continues through many pages.

A narration structure sometimes resembling a fugue or improvisation.

Intrigue articulated around the confrontation of specific characters, each in turn setting traps for the others.

Who would the patent holders have been? They could have been other novelists, perhaps Dumas or Balzac, who had written such novels - but not necessarily.

It isn't necessary to write a programme to patent a software idea, so if our hypothetical literary patents follow the real patent system, these patent holders would not have had to write novels, or stories, or anything - except patent applications.

Patent parasite companies - businesses that produce nothing except threats and lawsuits - are growing larger.

Given these broad patents, Hugo would not have reached the point of asking what patents might get him sued for using the character of Jean Valjean. He could not even have considered writing a novel of this kind.

This analogy can help non-programmers to see what software patents do. Software patents cover features, such as defining abbreviations in a word processor or natural order recalculation in a spreadsheet.

They cover algorithms that programmes need to use. They cover aspects of file formats, such as Microsoft's new formats for Word files. The MPEG 2 video format is covered by 39 different US patents.

Just as one novel could infringe many different literary patents at once, one programme can infringe many different patents at once. It is so much work to identify all the patents infringed by a large programme that only one such study has been done.

A 2004 study of Linux, the kernel of the GNU/Linux operating system, found that it infringed 283 different US software patents. That means each of these 283 different patents covers a computational process found somewhere in the thousands of pages of source code of Linux.

The text of the directive approved by the council of ministers clearly [authorises](#) patents covering software techniques.

Its backers claim the requirement for patents to have a "technical character" will exclude software patents - but it will not. It is easy to describe a computer programme in a "technical" way, the boards of appeal of the European Patent Office [said](#).

The board is aware that its comparatively broad interpretation of the term "invention" in Article 52 (1) EPC will include activities so familiar that their technical character tends to be overlooked, such as the act of writing using pen and paper.

Any usable software can be "loaded and executed in a computer, programmed computer network or other programmable apparatus" in order to do its job, which is the criterion in [article 5 \(2\)](#) of the directive for patents to prohibit even the publication of programmes.

The way to prevent software patents from bollixing software development is simple: don't authorise them. In the first reading, in 2003, the European parliament adopted the necessary amendments to exclude software patents, but the council of ministers reversed the decision.

Citizens of the EU should phone their MEPs without delay, urging them to sustain the parliament's previous decision in the second reading of the directive.

© 2005 Richard Stallman (rms@gnu.org). Verbatim copying and distribution of this entire article are permitted worldwide without royalty in any medium provided this notice is preserved.

• Richard Stallman launched the GNU operating system (www.gnu.org) in 1984 and founded the Free Software Foundation (fsf.org) in 1985. Gérald Sédrati-Dinet devised the examples in this article