Frequently Asked Questions about software patents

Protection of investment

Contents

'Isn't protection by patents necessary in order to protect one's investment against copying? There is no progress without such protection!'

Copyright is the perfect protection of investment for software.

Copyright has served well during the last twenty years as the motor of the software industry. With software it works even better than with books. One of the reasons is that in software a strict separation between editable source code and executable binary code is possible.

Software patents, on the other hand, are only used strategically in the current practice—i.e. in the USA. As investment protection they are much too inflexible, since they require waiting times ranging from six months to a few years and cost several tens of thousand of Euros. Copyright becomes effective automatically and immediately.

Maybe there are a few companies which could profit from software patents, but nobody can seriously argue that software patents are useful for the whole industry.

'If a developer has spent much time in an epoch-making algorithm, wouldn't it be appropriate to reward him with a patent?'

The developer can, with copyright as investment protection, transform his discovery into software. This has been working very well in the past.

A potential imitator who only knows the original as executable binary code, but not the source code, must spend the same amount of work as the original developer. On which economical or moral reason should he be prohibited from doing so?

The small inventor who obtains wealth from a patent and hard work is, by the way, nothing more than a nice fairy tale. In hard reality the biggest companies of the world are using patents in the hundreds as weapons against each other or against smaller and more flexible competitors.

'Then I take out a patent for a basic technique and become rich!'

This only works if you do not develop software yourself, but concentrate completely on patents. There are several companies living on a business model like this.

But as soon as you develop software yourself, you cannot avoid infringing on patents of large companies and will not be able to effectively prosecute them because of infringement on your own patents.

Current situation

'All the time I read “European software patents”. I thought they do not exist currently, but should only be allowed in future?'

That's not true. About 30,000 European software patents have been granted without a sufficient legal base. The current jurisprudence (“status quo”) differs very much from the current state of law (art. 52 EPC).

'The European patent offices and courts are not stupid and will be able to prevent trivial patents and defend the respectable entrepreneur against absurd claims.'

The European patent offices already have granted thousands and thousands of trivial patents, e.g. patent no. EP 394160 on the progress bar or patent no. DE10108564 on reception of e-mail; this against current law (Art. 52 EPC), which clearly prohibits software patents.

Even assuming European courts would have enough expert knowledge to see through a patent lawyer operating with expert vocabulary, in most cases it is sufficient to threaten legal action with a case value of millions of Euros in order to force a small or medium company or even a single developer to give up.

In the USA current legal cases impressively show where the development leads. Our only chance is to give no ground to those cases; this means retaining the current legal situation.

'Isn't the battle already won? The European Parliament rejected the software patents directive in July 2005.'

There are new attempts to introduce software patents, such as the Community Patent.

The Community Patent is the idea that inventions will only be patented in the EU, instead of having to register a patent in the patent office of each EU country. This seems like a good idea, but the problem is in the details. Current plans don't even mention the subject of software, although software patentability is one of the main drivers of these plans. Instead of directly imposing software patentability, the proposal is now to remove the patent system even further from legislative review by any democratically elected parliament. Thus in effect legislative power is handed over to a few top judges and to the circle of administrative officials that is running the European Patent Office and the EU Council's patent policy working

party. There are even moves to explicitly make EPO case law binding on the new EU patent institutions. Of course all this goes without mentioning the word "software" or "computer", but the underlying issue is clearly understood.

The Community Patent has failed for 25 years due to resistance from many quarters within the patent lobby itself. If now suddenly this resistance can be overcome, there isn't much need to explain what is the driving force that is overcoming it.

Trivial patents

'Wouldn't it be better to solve the problem of trivial patents by demanding a minimum level of invention instead of not accepting software patents in general?'

All recent experiences clearly show that this will not work.

The software patents already granted in Europe strongly show that patent offices are not able to prevent trivial patents. So the often stated better quality of European patent examinations is pure fiction.

'Aren't trivial patents an exception?'

No, they're the rule. If you have experience in software programming, it should be easy for you to convince yourself. The FFII has collected and documented about 10,000 of about 30,000 European software patents, available via http://eupat.ffii.org/patente/. Randomly pick any of these patents, read and understand the claims and judge for yourself:

- How large do you think is the effort for getting from the problem to the patented solution idea, in comparison to the effort for reading the patent application document?
- How high do you think is the probability that a programmer could accidentally violate a patent not known to him?
- If a customer would ask you to solve exactly this problem, how probable would you think it was that your independently developed solution would violate this patent?

For a small number of those patents we have an easy understandable short explanation; see http://eupat.ffii.org/patents/samples/.

One of the original aims of the patent system is the documentation of knowledge in patent applications. At this point it should be noted that this kind of “documentation of knowledge” in patents is completely useless for the programmer.

Software patents and Free Software

'Wouldn't it be better to vote for an exception for free software instead of trying to prevent software patents altogether?'

Such an exception would be the same as prohibition of software patents in general, since free software may be commercial software as well. The proponents of software patents know this and try to exploit this misunderstanding so that there will be at most an exception for non-commercial software. With this, nothing would be won, since a patent only claims commercial use of an idea. The possibility of commercial use is an important point of free software.

'Some free-software-projects are developed non-commercially. Could they be attacked by software patents at all?'

Yes. The patent owner can claim that the existence of this free software hurts him commercially.

Especially in the case of a non-commercial development, the mere threat of a lawsuit is often sufficient to force the developers to abandon the project. This is because there are no monetary means to finance the lawsuit.

'Can software which is distributed as source code be attacked by patents at all? (“source code privilege”)'

According to an early draft of the now rejected software partents directive, software could be attacked only from the moment when it would be executed on a computer; thus, not the author, but the customer would be liable. This wouldn't help me as the author, however, since my customer would hold me liable for patent claims by third parties.

The later draft of the European Council contained a new article, by which publication of source code could already be a direct infringement.

'If software patents are so dangerous for free software, why does free software also exist and grow in countries which have software patents?'

The great success of Free Software easily makes one oversee the damage already done by software patents. Some of the projects which had to be given up due to software patents were Free Software.

As long as software patents in Europe offically do not exist, many patent holders abstain from charges, because a wave of legal cases would heat up the debate about European software patents.

Proposals for a Solution

'Wouldn't it be a useful compromise to grant software patents for five years only?'

A shorter patent duration would of course shorten the duration of damage.

But this is not allowed by international law: the TRIPS agreement demands a minimum duration of patents of at least 20 years.

'Isn't is possible to work around software patents and use alternative methods? For example Ogg/Vorbis instead of MP3?'

In some cases it is really possible. The Ogg/Vorbis developers have done patent research and hope their format won't violate patents in the USA. On the other side, there are many areas where patents are so central and broad that working around them is impossible (e.g. panorama images).

But you can never be sure: patent research is not reliable. Even JPEG was believed for many years not to be covered by any patents. Now courts have to decide whether this is indeed true.

At least it is always a competitive drawback if you have to work around a file format which has been established as a de facto standard. Especially in the software sector, interoperability is very important.

'What should happen instead?'

Since software patents have been proven to have a negative impact on the economy they should not be granted at all.

A revision of the patent laws should make this clear. In practice a more narrow definition of the word “technical” is necessary.

Frequently Asked Questions about software patents (last edited 2014-05-11 09:30:37 by mages)