

# Collective Shields against Software Patents?

<http://swpat.ffii.org/analysis/shield/swpatgacri.en.html>

Workgroup

swpatag@ffii.org

english version 2003/12/16 by PILCH Hartmut\*

2004-02-04

Building a patent portfolio is hardly a realistic option for the self-defense of the free/opensource software community. However this community has a certain chance of leveraging prior art to its advantage, especially if it does this in a cost-efficient manner and realises the potential of legal insecurity inherent in the software patent system. So called Defensive Publishing, as has been proposed by initiatives like the Foresight Institute, however appears to be an incredibly silly approach. The most urgent task at this moment (spring 2001) is to time-stamp every CVS archive and every mailing list on which software ideas are developed.

## Contents

<b>1</b>	<b>Bad chances in defensive patenting</b>	<b>2</b>
<b>2</b>	<b>Good chances in prior art warfare</b>	<b>2</b>
<b>3</b>	<b>“Defensive Publishing” No Thank You!</b>	<b>3</b>
<b>4</b>	<b>Timestamping proprietary development also helps</b>	<b>5</b>
<b>5</b>	<b>What needs to be done</b>	<b>5</b>
<b>6</b>	<b>Further Reading</b>	<b>6</b>

---

\*<http://www.ffii.org/phm>

## 1 Bad chances in defensive patenting

Obtaining and maintaining patents implies high costs:

**cost of motivation:** Developers find new ideas every day and are usually not interested in the tedious work of distilling these ideas into patent applications. They want to develop products and not patents, and often resent patenting on various grounds. Even though some laws guarantee them a share in revenues from the patent, this usually is not enough of a motivation. Large companies like Siemens, IBM or SAP have therefore installed well-staffed committees of patent application experts and ample bonuses for developers in order to overcome this problem and provide sufficient incentives for developing patents.

**cost of obtaining:** According to EPO figures, a European patent costs an average of 30000 EUR in office fees, assuming that it is upheld for only 10 years and not translated to all European languages. Costs for patent attorneys are much higher and often calculated on a basis of 300 EUR per hour.

**cost of maintaining:** Even simple litigation against one infringer or opponent will cost more than 100000 EUR/USD in lawyer fees. Often the cost is in the range of millions, and many inventors even of important basic patents have gone bankrupt over this (e.g. Goodyear). Insurance policies for patent litigation either don't exist or are astronomically high.

These costs have to be earned back by exercising monopoly rights. This usually usually means a combination of upfront payment (lumpsum) and per-copy license fees. Free software and per-copy licensing are incompatible. Thus the patenting costs must be earned back by the upfront payment, e.g. 1 million USD in the case of MP3. There is usually no chance of receiving 1 million from anyone in return for a permission to write free software. There may however be a chance of gaining some money from vendors of proprietary software. However this income will at best be slim if you grant a free license to the competition from free software. Under these circumstances, the chances for using patents for defending free software are very poor if not non-existent.

## 2 Good chances in prior art warfare

We saw that the proprietary nature of the patent game makes it all but impossible for free software developers to organise any self-defensive effort within this game. However this same proprietary nature of the patent system generates weaknesses that can be exploited. Free software development produces the largest available body of published prior art. Such ideas are then in the public domain and any patents obtained thereon later are invalidated, once the prior art is found.

Strictly speaking this does not prevent anybody from asserting a trivial and broad patent for which no prior art exists, thus changing nothing about the most important source of danger. But given the difficulty of finding prior art and the uncertainty about whether any patent is really valid, a vast arsenal of hard-to-search prior art can be leveraged as an indirect protection shield for the whole free software community:

1. A large body of prior art that is published every day at a huge number of disparate locations all over the world
2. An inherent legal insecurity in the software patent portfolios of major players, caused by the impossibility of making prior art reliably searchable
3. A large community of highly competent and spirited software developers, capable of concerted action against those who attack free software.

Under these circumstances the most urgent thing to do at the moment is to automate time-stamping and provide it as a basic service in all major archives (CVS, Mailing Lists, HTTP etc) where free software and related discussions evolve. Wherever this kind of service is not available, developers should shift relevant discussions and publications to Usenet, since this is fairly reliably time-stamped by DejaNews.

### **3 “Defensive Publishing” No Thank You!**

The silliest thing for free software developers to do at the moment would help volunteer in helping patent offices to build prior art databases “in order to prevent bogus patents from issuing”, as has been proposed by the “Foresight Institutes” with the endorsement by some “top opensource leaders”. This so called “defensive publishing” is a boomerang both for the developer who attempts it and to the community. Let us explain this by comparing the “defensive publication” strategy to the normal decentral publication of prior art.

Perfect Defensive Publishing	Time-Stamping of the normal open development process
Patent-relevant ideas easy to research	Patent-relevant ideas hard to find.
Great effort and costs for the developer, lesser effort for the researcher and patent applicant	Less effort and costs for the developer, greater effort and costs for the searcher
Lower patenting costs, more (broad and trivial) software patents	Higher patenting costs, less (broad and trivial) patents
Your software idea will probably not be patented by anybody	Your software idea may be patented by someone, but that patent owner will avoid all conflicts with you and other free software developers as soon as you show him your proof of prior art. He will become very cautious even with his other patents, especially toward free software developers, because they can do him a lot of uncalculable harm. He will prefer to keep some invalid patents so as to placate his investors and raise funds, rather than really risk offending anyone. In some cases he will even pay you to keep silent about your prior art findings.
A lot of fortune-seekers will gather around the defensive-publishing database and in order to see what ideas could be patent-relevant and then see if they can find some complementary ideas for themselves to patent. These patents will then probably be valid. Your field of work will be blocked by a patent thicket.	No fortune-seeker will find your prior art information in time and you will keep a head-start, having good chances to step by step make a whole field of ideas patent-free.
Patent offices may look credible when they say: "Look, we have done our part to search prior art. The free software developers should stop whining and engage in defensive-publishing. If they do their job of building prior-art databases, then we will do ours of guaranteeing that their ideas aren't patented."	Patent offices may have to admit that the problem of harnessing prior art in software is not "just a matter of time".

## 4 Timestamping proprietary development also helps

Developpers of proprietary software should also make their work reconstructable day by day, using the same methods as developpers of free software.

Such a proprietary development history archive does not constitute “prior art” and can therefore not be used to directly invalidate the patent. But it can establish “prior use right”, meaning that the developer can continue to “sell products based on his idea”. He may however not transfer this right to third parties.

So far it is not completely clear what happens to prior use right when the owner of such a right decides one day to publish his formerly proprietary software under an opensource license. Does that constitute a transfer of the right to “sell products”? Or shouldn’t any prior use right owner be free to “sell” his “product” to his “customers” in any way he likes, based on the principle that his customers do not have to acquire separate usage licenses? It can be seen here that the assumptions underlying the patent system do not match the reality of information products.

However as long as there are no court verdicts to the opposite effect, we have good reasons to hope that the prior user is free to distribute his work as free software. In that case everyone else is free to circumvent the patent as long as he again lets his development work flow back into this free software.

Thus the prior user of a patented software idea has a particularly strong position. He can either make the idea free for everybody or negotiate a reward from the patentee for not doing so. For the public, prior use right has about half the value of real prior art. This half-value makes it worthwhile for developpers of proprietary software to make their development history archives accessible to the public after a certain period of time. This game, if played effectively, can significantly reduce the incentive to patent trivial innovations for which a lot of private prior use can be expected to exist even if no prior art is found.

## 5 What needs to be done

- Time-stamping of all development sites; lots of exhibitionist smalltalk about possible software ideas, preferably in various national languages
- Easy-to-understand public documentation of those software patents whose owners or licensees exploit them in a particularly abusive manner.
- Formation of cooperatives for the exploitation of prior art and for the acquirement of some useful defensive patents as far as that is possible.
- Continuously influence the public and the politicians in order to bring patent inflation under control and outlaw software patents.

## 6 Further Reading

- **Diskussion**<sup>1</sup>
- **Skeptical Article in Linux Week**<sup>2</sup>
- **Prior Art DataBase**<sup>3</sup>

A well-meaning project that may do more harm than good to free software

- **Software Patents Institute**<sup>4</sup>

An organisation founded by IBM and others with the mission of creating prior art databases for software (and defuse opposition to software patentability) in 1994. Apparently only the second aim was achieved.

---

<sup>1</sup><http://lists.ffii.org/archive/emails/swpat/2001/May/0064.html>

<sup>2</sup><http://lwn.net/2001/0510/>

<sup>3</sup><http://savannah.nongnu.org/projects/padb>

<sup>4</sup><http://www.spi.org/>