

Guide to Submissions on Innovation Patents

Ben Sturmfels

September 26, 2013

Dear Fellow Member of the Australian Software Industry,

Chances like this to eliminate the threat of software patents are rare. The Government is requesting submissions for a Review of the Innovation Patent System. **Please make a written submission by 4 October 2013.** I've prepared the following no-nonsense guide to help you.

If you have any questions, please call me on (03) 9024 2467. I'm happy to explain the issues without the legal jargon.

Yours sincerely,

Ben Sturmfels

1 What is our goal?

We have a perfect opportunity to eliminate the threat of computational idea patents (aka. software patents) in the Innovation Patents system. This win would be a significant milestone in our ongoing campaign to against such patents in the Standard Patent system.

The ACIP review has shown genuine interest in the problems around patents and computing and made detailed references to it in their report. We need your help to convince them to take the next step.

2 Who will read my submission?

The ACIP is a government-appointed council of experts with various backgrounds such as business, research, legal and academia. They are thoughtful and considerate people who have a genuine interest in improving our laws.

Please use your own words as much as possible and be brief and respectful.

3 What should I say?

1. State that you're concerned about "computational idea patents" and go on to define this (see Definition below). The ACIP expressed confusion, so we need to be very clear.
2. Mention you're a member of the Australian software industry. For maximum effect, please focus your comments on the **software industry only**.

3. State that both Standard and Innovation Patents are harmful to the software industry.
4. State that such patents don't encourage innovation in the software industry and, in fact, actively discourage innovation. It's probably best not to go into great detail about the reasons, but brief treatment is useful (see Reasons below).
5. For strategic reasons, avoid recommending tweaks to the patent system. Tweaks will never solve the fundamental problem that patents are a bad fit for computing, so are a distraction from our real goal.
6. Recommend that for both Standard and Innovation Patents, we legislate that developing, distributing, or running a program on generally used computing hardware does not constitute patent infringement. Follow up by saying that the alternative approach of excluding computational ideas from patentable subject matter would also be acceptable, but may be less effective (see Recommendations below).

Encourage your colleagues and your organisation's executive to co-sign your submission.

Post or email your submission to ACIP and BCC to ben@sturm.com.au.

Jeff Carl
Secretariat
Advisory Council on Intellectual Property
PO Box 200
Woden ACT 2606
mail.acip@ipaustalia.gov.au

4 Background

4.1 What is a patent?

Patents are a government incentive scheme to encourage people to invent things and publish details for the benefit of society. The patent owner is granted a monopoly for some time and the public receives information about how the invention works. Unfortunately this incentive scheme doesn't work as intended for the software industry, causing more harm than good.

4.2 Definition of "software patent"

The term "software patent" is misleading for us and for ACIP. A patent gives a monopoly not over a specific piece of software, but over the **ideas** used in the software. Software written in any programming language (even a language not yet invented) can infringe by implementing a patented idea.

For this reason, we prefer terms like "computational idea patents" or "patents on computation and information processing". These terms express that we're talking about the methods (aka. algorithms) used in the program, not the program code itself.

4.3 Reasons why these patents are harmful

The patent system was designed before the software industry existed and is unsuitable, ineffective and indeed harmful to this industry:

1. **Making software is faster and cheaper than other industries.** You don't need a large expensive lab for research. You don't need a manufacturing plant to publish. Our components don't overheat or wear out.
2. **Software systems are comparatively more complex.** Being faster and cheaper to build, we can put hundreds of ideas into a single program as opposed to products in slower moving industries.
3. **Less overhead means more people get involved.** All over the world, 14-year-olds in their bedrooms have access to industry standard tools.
4. **Independent invention is very common** because so many people are working on the same problems. The patent system handles this aspect very unfairly in that everyone other than the first person to patent is infringing.
5. **Reuse of ideas is best practise.** Software simply can't exist without reusing ideas that other people have come up with. Re-inventing from zero is simply not possible.
6. **These patents cover ideas, not implementations.** Due to the nature, patents on computational ideas have a much broader reach (more damaging) than the fairly narrowly defined scope of physical inventions.
7. **8 years is a lifetime in computing.** Innovation Patents cover up to 8 years, Standard Patents up to 20 years. This is long enough to render an idea useless to the industry.
8. **We were doing fine before 1991** when the first Standard Patent on a computational idea was granted in 1991. The software industry has had a long history of rapid innovation.
9. **After 1991, innovation continues merrily without relying on patents.** Examples of innovative Australian software without patents include Trumpet Winsock (networking, 1993), rsync (data synchronisation, 1996) and netfilter/iptables (firewall, 1998).
10. **Computing is interesting, challenging and profitable.** Further incentive is simply not required.

2008 Venturous Australia Report by Dr Terry Cutler and a panel of independent experts said: "... particularly in new areas of patenting such as **software** and business methods, there is strong evidence that existing intellectual property arrangements are **hampering innovation.**"

4.4 Recommendations in detail

Experts are now suggesting a strategic deviation in our campaign to focus on limiting the effect of computational idea patents rather than making the subject matter non-patentable. It's solving the problem at the other end of the system, shielding us from infringing rather than trying to eliminate the patents themselves.

This is a good strategic move because:

- It doesn't require classifying patents or patent applications as "software" or "not software."
- It provides developers and users with protection from both existing and potential future computational idea patents.
- Patent lawyers can't defeat the intended effect by writing applications differently.

4.5 What is an Innovation Patent

An Innovation Patent gives an inventor up to 8 years monopoly over an idea. By comparison, a Standard Patent provides up to 20 years monopoly over an idea. These could be thought of as "light" versions of a Standard Patent. This approach is similar to the "Utility Model" used in some countries. The inventiveness requirements are much less and they're faster and cheaper to register. The patent isn't actually examined and certified unless required (eg. someone wants to sue over it). The Innovation Patent system came into effect late 2000 replacing the petty patents system.

Innovation Patents do cause unique problem by their nature, but thankfully there appears to be general consensus across industries about their shortcomings. There's no need for us to fight specifically on the issues of Innovation Patents, as other people are doing that for us. We can focus our energy on opting out of the whole system.

4.6 Background on the review

The Review of the Innovation Patent System commenced in 2011. Having missed opportunities before, many in the software industry realised that this was a great chance to have a say, so made submissions to this first round.

Reflecting on this, ACIP have suggested some steps forward and are asking for further comment in this second round. From speaking with ACIP, it turns out that they just didn't get enough engagement in the first round, so the second round is actually a bit of a reboot of the review, branded as a "follow-up".

Happily though, our first round comments had a substantial effect on ACIP. Here are a few notes from their Options Paper:

"Terms of Reference . . . Inquire, report and make recommendations to the Australian Government on the effectiveness of the innovation patent system in stimulating innovation by Australian small to medium business enterprises . . . any unintended consequences arising from its implementation."

"Half of the submissions (17) were from workers in the software industry supporting an abolition of computer-related patents. . . ."

"It is thus not possible using the results of this study to calculate the net effects of innovation patents on Australia's level of innovation."

"Information technology" accounted for 14% of innovation patents in 2011 and 12% in 2012.

"Apple is the biggest single user of the innovation patents system and owns about six percent of all innovation patents certified since 2006." (That's 98 patents which is more than twice the next largest portfolio of Aristocrat Technologies with 43.)

“An overwhelming number of comments received . . . supported the exclusion of software.”

“ACIP notes that Japan and the Republic of Korea already exclude computer software from their utility model systems.” (The New Zealand legislation applies to Standard Patents not Utility/Innovation Patents.)

4.7 About this campaign

I’m Ben Sturmfels, a software developer and small business owner. I don’t hold any patents and haven’t (yet) been threatened or sued over patents, but for a long time have been concerned by this threat to software I rely heavily on.

For the past few years I’ve been spending a little of my spare time directing a campaign to eliminate the threat of software patents in Australia. This campaign is called End Software Patents Australia and is heavily influenced by the hard work of Ciarán O’Riordan, director of the international End Software Patents campaign.