

A note on open source, public domain source and innovation

Often in discussions about open source and innovation the debate quickly turns to a discussion about the relationship between proprietary code and open source – often loosely defined as software licensed according to OSI-accredited licenses. This narrows the debate in an unfortunate way. Arguments exist on both sides, and the debate rages on (as it should).¹ There is another question, however, that would be interesting to pursue in this field, and that is the question of open source versus public domain. One question, in short, would be: does open source software foster innovation better than software available in the public domain?

The question might seem a strange one, since the public domain and open source are often treated as synonymous or at least closely related. There are, however, small differences. In reality, open source software and proprietary software actually, and this comes as no news to most people, have much in common: both are governed by licenses, using them has legal effects and they are regulated by intellectual property rights.

The wikipedia definition of free software makes this point nicely:²

Free software, as defined by the Free Software Foundation, is software which can be used, copied, studied, modified and redistributed with little or no restriction beyond the requirement that source code must be made available.[1] Freedom from such restrictions is central to the concept, with the opposite of free software being referred to as proprietary software by the Free Software Foundation and others in the free software movement (a distinction unrelated to whether a fee is charged). The usual way for software to be distributed as free software is for the software to be licensed to the recipient with a free software license (or be in the public domain), and the source code of the software to be made available (for a compiled language).

Free software, here, includes – in passing – public domain software, but focuses on licensing. Open source software – even more clearly – is defined as software adhering to certain licenses.³ But what of software that is not governed by licenses, that can be

¹ The pro argument being at least that some kind of patent rights seem to foster innovation. See Mann, R J Do Patents Facilitate Financing in the Software Industry?, 83 TEXAS LAW REVIEW 961 (2005) (available online at http://www.utexas.edu/law/faculty/rmann/info/Data/Software_Patents.pdf). See, however, also Bessen, J A Comment on “Do Patents Facilitate Financing in the Software Industry?” (available online at <http://www.researchoninnovation.org/comment%20on%20Mann.pdf>). The counter-argument being that innovation is best fostered without patents or other limiting rights, see for example Bessen, J et al *An Empirical Look at Software Patents* (Federal Reserve Bank of Phila.) (available online at <http://www.researchoninnovation.org/swpat.pdf>) arguing the software patents actually substitute for investments in research and development and that they thus suppress innovation.

² Quoted 2007-02-07 and available at http://en.wikipedia.org/wiki/Free_software

³ See for example the definitions at the Open Source Initiative (available online at <http://www.opensource.org/docs/definition.php>).

freely used and is exempt from intellectual property rights? What kind of software would that be?

Firstly it would be software that has passed into the public domain and cannot be protected anymore: software where the intellectual property protection has expired. One reason why we do not discuss public domain software much is that the software industry still is young and it can be discussed how much software there is that has had intellectual property protection that no longer applies.

The other example would be software that either is not eligible for intellectual property protection at all or where the user has found a way to permanently exempt it from intellectual property regimes. The latter – as we will see – is hard to do.

The former, however, is simple: here we discuss small pieces of software that are openly available to all, but not of such sophistication as to be eligible for copyright or patent protection. These “snippets” of code are openly shared everywhere on the internet in different languages: javascript, php, java, python and many other.

To actually exempt code that could be copyrighted or patented from intellectual property protection turns out to be surprisingly hard. To simply say that one has no interest in the code and that it can be distributed freely might not be enough. At least in Sweden copyright will be awarded automatically to anybody without any tests and the end-result will be that as a creator of code you will have copyright whether you want it or not. Now, you may state that you have no intentions of exercising those rights, but can you actually rid yourself of them?

The answer seems to be no. There is no simple legal procedure to terminate copyright, at least. Indeed, once a patent has been awarded there seems to be no simple procedure to negate it either. In patenting, however, there is an interesting possibility to choose between publishing – establishing prior art – and patenting. Publishing would destroy for any future party that has plans on patenting the same thing, and not generate anything else than an intellectual property right in the form of copyright in the published paper.

Now, a number of interesting questions can be asked, and should be asked in the debate on innovation and software.

- 1) How much of the innovative value in what is usually attributed to open source is actually public domain source consisting of snippets and simple sets of code that are not eligible for any intellectual property rights?
- 2) What are the differences between public domain source and open source in terms of fostering innovation?
- 3) How could new legal instruments be crafted to allow irreversible termination of patents and copyrights?

Let me try to sketch answers to these three – admittedly large – questions.

First, I think it can safely be noted that the first question is the hardest one. To try to differentiate between the innovative value of public domain software and open source software is inherently very difficult, and thus something that I think deserves more space than I will afford it here. Some small points can be made, however. Initially it depends on the question of what counts as public domain software. What is – in short – a “snippet”? Where does the boundary go for software to be eligible for intellectual property rights? It seems obvious that this is a moving target. The basic program “10

print “nicklas” 20 goto 10” seems to be well below. But what about this snippet for creating pull-down menus in javascript?⁴

<!-- TWO STEPS TO INSTALL PULLDOWN MENU:

1. Paste the coding into the HEAD of your HTML document
2. Add the last code into the BODY of your HTML document -->

<!-- STEP ONE: Copy this code into the HEAD of your HTML document -->

<HEAD>

<SCRIPT LANGUAGE="JavaScript">

<!-- Original: Alex Tu <boudhal@hotmail.com> -->

<!-- Web Site: http://www.geocities.com/alex_2106 -->

<!-- This script and many more are available free online at -->

<!-- The JavaScript Source!! <http://javascript.internet.com> -->

<!-- Begin

```
function formHandler(form){
```

```
var URL = document.form.site.options[document.form.site.selectedIndex].value;
```

```
window.location.href = URL;
```

```
}
```

```
// End -->
```

</SCRIPT>

</HEAD>

<!-- STEP TWO: Paste this code into the BODY of your HTML document -->

<BODY>

<center>

<form name="form">

<select name="site" size=1>

<option value="">Go to...

<option value="http://www.yahoo.com">Yahoo

<option value="http://www.metacrawler.com">Metacrawler

<option value="http://www.altavista.digital.com">Altavista

<option value="http://www.webcrawler.com">Webcrawler

<option value="http://www.lycos.com">Lycos

<option value="http://javascript.internet.com">JavaScript Source

</select>

<input type=button value="Go!" onClick="javascript:formHandler(this)">

</form>

</center>

<p><center>

⁴ From <http://javascript.internet.com/navigation/menu.html>

Free JavaScripts provided

by The JavaScript Source
</center><p>
<!-- Script Size: 1.20 KB -->

The question is not trivial. I would surmise that this too falls below the threshold for copyright protection – but I am not sure. Especially not since it contains instructions on implementation as well. If we assume a percentage of what is commonly described as open source is actually not eligible for copyright protection at all, we find something else that is surprising, and that is that some of it may still be governed by licenses. By blindly focusing on “open source” instead of the public domain many software makers may lock in software in licenses that would actually not be eligible for IPR-protection, thus creating a set of code that could be public domain, but now is controlled by agreements and licenses instead. Open source may thus actually hamper the development of the public domain. This being said I think one could safely say that the innovative effect of public domain software is not non-negligible. In fact, I would argue, that much of innovation is combining different pieces to create something new. The smaller the pieces are and the more generally they are designed the larger the chance that they can contribute to innovation *and the chance that they are not eligible for IPR-protection*. It would therefore be possible to guess that a large share of the innovation that is commonly attributed to open source software actually should be attributed to public domain software. (Indeed, it may even be possible to argue that open source software depends on public domain software to some degree.)

The second question is also quite hard, and it can actually be interpreted two ways. The first would be focusing on the existing set of public domain software, the “snippet”-factor if you will, and the second would be asking how a large public domain software sector would impact on innovation. This question, in turn, can be asked in relation to proprietary software or to open source software, or the two bundled as “controlled software”. One possible answer (with weaknesses, I’ll admit) would be to compare this question to asking what the impact of a cultural heritage is on literary creativity. If we look at other sectors with large sets of public domain content, we may evaluate the impact it has on music, literature, art and engineering. It would seem that the impact of the public domain on creativity is several degrees larger than that of works that are merely licensed freely. But this is a mere sketch of an answer.

The third question is the perhaps most operational. It should be possible to argue that we need a mechanism of termination, a negapatent of some kind that if applied for could effectively terminate any intellectual property rights – copyright, patents and all such - awarded. Such a negapatent would have intriguing economic effects on the IPR-market, especially for software companies. Many software companies today practice defensive patenting and accrue legal costs that are quite daunting. If the alternative is to negapatent something, thus voiding any chance that the competitors can acquire *any* rights to similar solutions, this would probably be an attractive economic alternative (or it could be if designed well). Any mechanism for terminating intellectual property rights and irreversibly placing something in the public domain would be worth examining.

In summary, then, it seems that the dichotomy between open source and proprietary software is a false one. We need to seriously consider a third option for openness and evaluate its impact on innovation.