By Paul Belleflamme, 14 October 2012

A shorter patent term for some innovations?

Source: technorati.com

In a recent article (October 7, 2012), entitled, "The patent, used as a sword", the New York Times gives a critical view of the current state of the US patent system:

> *Patents are vitally important to protecting intellectual property. Plenty of creativity occurs within the technology industry, and without patents, executives say they could never justify spending fortunes on new products. And academics say that some aspects of the patent system, like protections for pharmaceuticals, often function smoothly.*
>
> *However, many people argue that the nation's patent rules, intended for a mechanical world, are inadequate in today's digital marketplace. Unlike patents for new drug formulas, patents on software often effectively grant ownership of concepts, rather than tangible creations. Today, the patent office routinely approves patents that describe vague algorithms or business methods, like a software system for calculating online prices, without patent examiners demanding specifics about how those calculations occur or how the software operates.*
>
> *As a result, some patents are so broad that they allow patent holders to claim sweeping ownership of seemingly unrelated products built by others. Often, companies are sued for violating patents they never knew existed or never dreamed might apply to their creations, at a cost shouldered by consumers in the form of higher prices and fewer choices.*

The extent to which software should be patented has been – and still is – a topic of intense debate. Nowadays, there is no harmonization across countries on this matter. As indicated in the quote above, the US Patent and Trademark Office broadly grants patents that may be referred to as software patents (and it has been doing so since at least the early 1970s). In contrast, the

European Patent Office has adopted a more restrictive approach (see here):

> *Programs for computers as such are excluded from patentability (…) a program for a computer is not patentable if it does not have the potential to cause a "further technical effect" which must go beyond the inherent technical interactions between hardware and software.*
>
> *(…) A patent application for an Internet auction system was not granted because the system used conventional computer technology and computer networks – which meant it made no inventive technical contribution to the level of existing technology. Such a system may provide business advancement to its users, but that is not the type of advancement required by the EPO.*
>
> *On the flip side, the problem of improving signal strengths between mobile phones is a technical problem, even if it is solved by modifications to the phone software rather than its hardware. Such an invention would obtain a patent, provided that the solution is also novel and inventive.*

The contrasting approaches of the USPTO and the EPO are in terms of "patentability": what are the conditions for a software (or a "computer-implemented invention" in the European terminology) to be patentable? Another, and potentially complementary, way to tackle the issue would be to discuss the "patent term" for software. Some scholars, among them Richard Posner (Federal appeals court judge and professor at at the University of Chicago Law School), recommend a shorter patent term for digital technologies.

How long should software patents last? Although Posner does not propose an explicit length, others do. For instance, according to the Electronic Frontier Foundation,

> *A patent covering software should survive for a term of five years, beginning from the date the application is filed.*

And you? What do you think? If you had to weigh the pros and cons of software patents, what patent duration (between 0 and 20 years) would you recommend? To help you organize your thoughts, you can use the framework proposed by William Nordhaus in his book "*Innovation, Growth, and Welfare*" published in 1969 (for a presentation of this framework, see the slides for Lecture 4).