

Why are Software Patents so Trivial?

<http://swpat.ffii.org/analysis/trivial/swpatfrili.en.html>

Workgroup

swpatag@ffii.org

2004-03-22

Most software patents are trivial and broad. The reasons are to be found in the patent system itself rather than in the insufficiencies of the examination process. On the one hand, the requirement of “non-obviousness” is not and can not be as performant as many “patent reform” proponents seem to wish. It can work only as one (weak) filter among several, which in combination serve to raise the overall score rate of the patent system, i.e. the ratio of “good patents”. The extension of patentability to software comes at the expense of several needed filters. The system has thereby been so badly compromised, that, no matter how the non-obviousness screw is turned, it is unlikely that its score ratio can be raised back to a level that could justify its continued existence.

Contents

1	A problem of law, not of patent examination	1
2	Why not just raise the “inventive step” standard?	3
3	The broken regulative: technical character	4
4	Further Reading	6

1 A problem of law, not of patent examination

By browsing through our European Software Patent Database¹, you can easily compile a “horror gallery” of impressively trivial and broad patents. It will be much more difficult to find even a single claim object that a programmer would find admirable and possibly worthy of a special exclusion right.

Some people say that this is because the patent offices do not do their job well. If the criteria of “novelty” and “inventive step” aka “non-obviousness” were applied 100% correctly, these people say, software patents wouldn’t do much harm. Some have been prophesying for a decade that is only a matter of time until this problem will be solved. However the solution seems further away than ever.

The following dialogue between Ramon Garcia Fernandez, a spanish information scientist, and Steve Probert, deputy director of the British Patent Office, conducted 2000-12-20 on a publicly archived mailing list², goes right to the heart of the problem:

Ferdandez: My disagreement with software patents is the philosophy of most patent offices. For me, a patent should be granted only where the research needed for the invention is expensive, so that said research would not be posible without the incentive of the monopoly. By contrast, the opinion of patent offices seems to be that the inventor should somehow have natural right to monopoly. They seem to think that the invention is intrinsically a very difficult process and that all incentive to it is small (I would like to hear Steve’s opinion). By contrast, the invention of a method to solve a problem is part of the daily work of a programmer.

Probert: I don’t really want to express an opinion because I think that would be wrong for me as a Civil Servant and an official of the Patent Office. What I would say is that the patent system has never differentiated between inventions on the basis of how much the underlying research cost. It doesn’t even distinguish between those inventions that are truly valuable and ground-breaking, and those that are [comparatively] trivial and insignificant. As the law stands, we would not be able to do so even if we wanted to. Only if an invention is known or obvious can we raise a legal objection. To some extent the system should be proportionate and self-regulating - in industries where the research costs are very high (eg pharmaceuticals), the patents will be more valuable. (So licencing costs will be higher etc.) OTOH, where the development costs are lower it will often (but not always, I accept) be cheaper to work around a patent. In such cases the patent will be worth less. In other words what I am saying is that the value of a twenty year monopoly varies depending on a number of factors, one of which is the typical cost of research in the field. Setting arbitrary thresholds based on research costs would effectively discriminate against individuals and smaller companies in

¹<http://swpat.ffii.org/patents/swpatpikta.en.html>

²<http://aful.org/wws/arc/patents/2000-12/msg00111.html>

favour of the bigger companies who will always spend more on research. And who would decide how much money needed to be spent to make a particular invention? (I presume you would not be content to determine the fate of a patent application on the basis of how much was actually spent.)

Fernandez: Probably the most important problem here is communication between non-programmers (lawyers, patent officials and so on) and programmers. It is probably very difficult to convince the former about what kind of things are easy for programmers, such as having ideas, and what kind of things are difficult and time consuming.

Probert: I cannot speak for lawyers, but I can assure you that many Patent Examiners are programmers themselves. In my group, all the Patent Examiners who deal with software applications either write computer programs in their spare time or have been employed as programmers before they became patent examiners. They usually have a pretty good idea whether something would have been easy or time consuming for a programmer. However, they might express the communication problem the other way around - it's very difficult to persuade programmers that just because an invention is "easy", does not make it any less patentable.

As can be learnt from this dialog, the non-obviousness criterion is not what the naive engineer or programmer thinks it is.

Criteria such as "novelty", "non-obviousness", "technicity", "industriality" may seem intuitively reasonable, and the naive patent professional may believe that they are the very yardsticks of reason.

The naiveté of both the engineer and the patent professional consists in forgetting that "the law is an ass", as the proverb goes.

The primary function of the "non-obviousness" criterion is not to assure what an engineer or programmer thinks is a reasonable "inventive height", but to provide a filter that can be applied by patent examiners in a predictable way.

Applicability and reasonability are two entirely different requirements, which need not not match.

It is enough if the "non-obviousness" filter helps, in combination with other filters, to yield a reasonable amount of "good patents" in the end. Patents will be deemed "good" if the scope of exclusion is felt to be

1. sufficiently broad to allow the patent owner to at least earn back the patenting costs (costs of patent acquisition and enforcement, disadvantage incurred by disclosure)
2. sufficiently narrow to avoid imposing unjustified costs on competitors (e.g. to make inadvertent infringement unlikely).

The "non-obviousness" filter is a means of raising the score rate of the patent system, i.e. the percentage of "good patents".

We have proposed means of estimating this score rate for various fields³. It seems to be well below 50% in most fields and below 1% when it comes to software and business methods.

2 Why not just raise the “inventive step” standard?

The Fernandez-Probert dialog illustrates what everybody in the patent trade knows: the criterion of “inventive step” as it stands is not designed to sort out trivial patents by itself, but to help, in combination with other filters, to raise the score rate of the patent system.

It is very difficult to prove that even the most trivial new idea does not contain an inventive step. The EPO’s Examination Guidelines of 2001⁴ even admonish examiners to be very critical of such proofs, apparently for good reasons:

It should be remembered that an invention which at first sight appears obvious might in fact involve an inventive step. Once a new idea has been formulated it can often be shown theoretically how it might be arrived at, starting from something known, by a series of apparently easy steps. The examiner should be wary of ex post facto analysis of this kind.

The EPO basically treats “inventive step” as an extension of “novelty”. In order to prove “lack of novelty”, an opponent must point to one single prior art document whose teaching falls into the scope of the claim. If no single document is found, the opponent will try to show that the person skilled in the art would have arrived at the teaching by combining two documents. In this case the claimed invention is said to be unpatentable due to “lack of inventive step”.

In the case of software patents, the person skilled in the arts rarely even consults documents. New programming problems occur all the time, and “inventing” a solution on the fly is the normal way to go. Most such solutions are not even published in any information science journal that an examiner might consult for testing their novelty. Rather, the program will usually also function as its own publication — a form of publication which can pose a severe challenge for the novelty concept of the patent system.

Yet not every innovation in the field of computing is produced on the fly, and some major advances are discussed in some highly-respected journals. So, shouldn’t we try to raise the inventivity standard, so that those really resepectable achievements can be singled out for rewarding by a patent (or a patent-like exclusion right) ?

Maybe. It sounds almost as tempting as “lasting world peace” and “real socialism”.

Even if lawmakers wanted to seriously tinker with the existing patent system and “raise the inventivity standard”, as has become a mantra in some “patent reform” proposals, they would encounter enormous practical difficulties:

³<http://swpat.ffii.org/analysis/testsuite/swpatmanri.en.html>

⁴http://www.european-patent-office.org/legal/gui_lines/e/c_iv_9.htm

fuzziness of “invention height”: How would they define “inventive height” (Erfindungshöhe), as it is still sometimes called in old-fashioned German patent jargon? “At least 20 inches above the state of the art”? “At least 20 twists of the brain (= 20 Lemelson ?) beyond the state of the art”?⁵

triviality by sequentiality: Abstract-logical constructions are usually composed of many small innovation steps that neatly build on each other to form a perfect whole. Each patent will usually focus on one of these logical steps, thus making this patent trivial and broad, even if the innovation itself was truly ingenious. This has happened e.g. in the case of the MP3 patents⁷. Even assuming it was feasible to narrow down the claim scope of the MP3 patents by applying “strong non-obviousness requirements”, this would then probably result in too narrow claims, i.e. claims that don’t allow the rightholder to earn back the patenting costs.

Incommensurate blocking effect due to non-triviality: In the (very rare) case of fundamental breakthroughs in abstraction (e.g. the Karmarkar inner-point optimisation method⁸), it may be impossible to define a claim scope that is both rewarding to the applicant and not too burdensome on follow-on innovation at the same time. This is the main reason why mathematics and discoveries have been excluded from patentability.

3 The broken regulative: technicity, industriality

Traditionally there has been one other criterion that has helped if not to sort out trivial ideas then at least to significantly lower the ratio of trivial patents: the requirement of “technicity” or “industriality”, which limits the patent system to applied natural science and matter-producing industries, more precisely defined as the *requirement that forces of nature be part of the problem solution which is rewarded by the patent*. In most countries of the world this requirement has in one or the other wordings been part of the patent law tradition at least until recently. The European Parliament has reaffirmed this requirement in September 2003 by voting⁹ for a strict definition of the concepts “invention”, “technical” and “industrial” along these lines. This criterion excludes those “post-industrial” innovations that are based only on abstract calculus and do not require experimentation. Finding a new causal relation between natural forces and a physical effect is usually much more costly than finding a new mathematical relation.

⁵It has been suggested⁶ that a social game could help here: let the patent applicant first publish only his problem and provide incentives for the public to submit solutions until a certain deadline. All these solutions are then considered to be prior art. This could really work, but it is a very radical reform proposal for a patent system which is governed by strong forces of inertia, who will always find forceful legal arguments against any even very moderate reform proposal that has a true potential of diminishing the number of patents granted.

⁷<http://swpat.ffii.org/patents/effects/mpeg/swxai-mpeg.en.html>

⁸<http://swpat.ffii.org/papers/konno95/konno95.ja.html>

⁹<http://swpat.ffii.org/papers/europarl0309/europarl0309.en.html>

While mathematical relations are composed of tiny functional elements that combine to a perfect whole, the physical world is causal rather than functional, and the whole is not the sum of the parts. Material phenomena may be described by mathematics, but such description is at best an approximation. Even a system of lego bricks will usually not work out the way you you may have built it in your mind. The more the system becomes complex, the more deviations can accumulate into unforeseeable effects. Undisturbed cleanrooms exist only in the world of ideas.

When the patent system is no longer limited to industrially applied natural science, its score rate plummets. Software is an art of abstraction and software patents come as a result of an opening of the patent system toward the abstract and functional, a proliferation of “function claims”, i.e. the patenting of unspecified “means” to achieve some so-called “technical effects”. This means that problems and not solutions are claimed. Since no new causal chain between material means and material results is involved, it becomes difficult for the patent applicant to claim his “invention”. It will usually be neither permissible nor economically rewarding to claim the specific mental steps by which a computing problem is solved. Rather, the patentee will try to claim the input and output (i.e. the “technical effects”) of the operation. However, unlike in chemical patents, there are no “surprising effects” to be claimed. Everybody knows that a computer can output calculation results to the screen. The only difficulty lies in knowing *how to tell the computer to do it*, and that is routinely left to the thousands of creative programmers, who, if allowed to do so, could independently devise hundreds of different creative solutions, all of which produce the claimed “technical effect”, but none or few of which are disclosed in the patent description. These and similar problems have been analysed in detail by some of the patent examiners who are struggling with them¹⁰. There is moreover a literature of mathematicians and epistemologists who explain in detail why the models are broken¹⁴ when the requirement of technical character (concreteness and physical substance) is given up. The German Federal Court of Justice already warned of this in the concluding remarks of its Disposition Program decision¹⁶ of 1976, which laid the foundations for the non-patentability of software in Germany:

However in all cases the plan-conformant utilisation of controllable forces of nature has been named as an essential precondition for asserting the technical character of an invention. As shown above, the inclusion of human mental forces as such into the realm of the forces of nature, on whose utilisation in creating an innovation the technical character of that innovation is founded, would lead to the consequence that virtually all results of human mental activity, as far as they constitute an instruction for plan-conformant action and are causally overseeable, would have to be attributed a technical meaning. In doing so, we would however de facto give up the concept of the

¹⁰see Dr. Swen Kiesewetter - Köbinger 2000: Über die Patentprüfung von Programmen für Datenverarbeitungsanlagen¹¹, Softwarepatente ohne Grenzen¹² and Dr. Swen Kiesewetter-Köbinger: Stellungnahme zur Patentierbarkeit von Softwarekonzepten¹³

¹⁴see Tamai 1998: Abstraction orientated property of software and its relation to patentability¹⁵

¹⁶<http://swpat.ffii.org/papers/bgh-dispo76/bgh-dispo76.en.html>

technical invention and extend the patent system to a vast field of achievements of the human mind whose essence and limits can neither be recognized nor overseen.

[...]

[...] from a purely objective point of view the concept of technical character seems to be the only usable criterion for delimiting inventions against other human mental achievements, for which patent protection is neither intended nor appropriate. If we gave up this delimitation, there would for example no longer be a secure possibility of distinguishing patentable achievements from achievements, for which the legislator has provided other means of protection, especially copyright protection. The system of German industrial property and copyright protection is however founded upon the basic assumption that for specific kinds of mental achievements different specially adapted protection regulations are in force, and that overlappings between these different protection rights need to be excluded as far as possible. The patent system is also not conceived as a reception basin, in which all otherwise not legally privileged mental achievements should find protection. It was on the contrary conceived as a special law for the protection of a delimited sphere of mental achievements, namely the technical ones, and it has always been understood and applied in this way.

Any attempt to attain the protection of mental achievements by means of extending the limits of the technical invention – and thereby in fact giving up this concept – leads onto a forbidden path. We must therefore insist that a pure rule of organisation and calculation, whose sole relation to the realm of technology consists in its usability for the normal operation of a known computer, does not deserve patent protection. Whether it can be awarded protection under some other regime, e.g. copyright or competition law, is outside the scope of our discussion.

As foreseen by the Federal Court in 1976, the introduction of software patents has opened a can of worms. It has not only removed the only viable criteria for limiting the scope of patentable subject matter, but also broken the models of the patent system on which requirements such as novelty, non-obviousness and enabling disclosure are built. It has overturned the balance of the patent system, leaving it behind in a state of inconsistency and dysfunctionality.

The European Parliament's vote has given the patent community a chance to repair its system. By reintroducing the requirement of concreteness and physical substance (technical character), the score rate of the system can perhaps be brought back to acceptable levels. Due to the unwieldiness of the system, there may not be many such chances. If this chance is missed, people in a large majority of disciplines may soon be voicing doubts about the legitimacy of the whole system and pressing for radical reform in unforeseeable directions. Judging from the cyclical movements of the patent system during the last two centuries, it would not be surprising if a failure by the patent

institutions to seize their chance could open the way to the strongest anti-patent wave that history has seen so far.

4 Further Reading

- **Report on Software Patents by Joaquin Seoane (professor of the Universidad Politécnica de Madrid) and Ramon Garcia Fernandez¹⁷**

Contains a long chapter on the triviality problem. Quotes patent lawyers who are encouraging their customers to patent anything that seems useful and forget about the idea that patents are about inventions. They are not. Especially in software, you have to forget that old prejudice quickly, if you want to play the patent game.

- **Enlightening Advice of a Patent Lawyer to Software Companies¹⁸ (cf Claus Dendorfer 1998: Patente und der Schutz softwarebezogener Erfindungen¹⁹)**

A German patent attorney explains to his customers that it is fairly easy and worthwhile to obtain patents with broad claims to trivial software ideas in Germany: One of the reasons for this triviality is the approach of patent offices to “non-obviousness” which implies that anything that goes beyond the cognitive capabilities of a standardised uncreative search robot is not obvious. Therefore it is especially easy to obtain broad patents in new areas of programming where little has been documented so far. Traditionally, software applications were frequently rejected due to lack of technical character, but recent court decisions have changed this. Since patents can significantly contribute to the value of a company, an investment in the systematic acquisition of patents can easily become worthwhile for a software company.

- **Bronwyn H. Hall & Rose Marie Ham: The Patent Paradox Revisited²⁰**

Research work done at Univ. of California, Berkely, published 1999 by National Bureau of Economic Research Inc. Finds that the surge in patents in the semiconductor industry in the 1980-90s does not reflect a surge in R&D activity.

¹⁷http://www.dit.upm.es/joaquin/report_en.pdf

¹⁸<http://lists.ffii.org/archive/emails/swpat/2001/Jun/0224.html>

¹⁹http://www.patent.de/swp_d.htm

²⁰<http://swpat.ffii.org/papers/nber-hallham99/nber-hallham99.en.html>

- **Lester C. Thurow 1997: Needed: A New System of Intellectual Property Rights: Squeezing today's innovations into yesterday's system simply won't work**²¹
- **Newell 1986: The Models are Broken**²²

Paul Newell, professor of computer science, in response to law professor Donald Chisum's proposal to affirm the patentability of algorithms, does not directly say whether algorithms should be patentable, but rather describes how both affirmation and negation of this proposal would clash with the underlying assumptions of the patent system and how this will inevitably challenge the foundations of the patent system. As more and more problems are solved by purely mathematical means, the patent system will either have to become less relevant for innovation, or it will have to completely review its model of what an invention is and how it should be appropriated. In particular, Newell explains some basic concepts of informatics and points out that, with algorithms, there can be no meaningful differentiation between discovery and invention, between application and theory, between abstract and concrete, between numeric and symbolic etc. Moreover he explains by a model of game theory that sharing algorithms, as currently practised by programmers, may lead to more innovation than making them appropriatable, so that a crude application of the patent system to algorithms could very well stifle rather than stimulate innovation.

-

- **Gregory Aharonian: The Patent Examination System is Intellectually Corrupt**²³

The author's indignation with the patent examination process leads him to sharply analyse many present-day problems. Whether these problems can, as Aharonian suggests, be solved by a mere improvement of the examination process, seems questionable. The indignation may be a result of wrong expectations due to the author's firm belief in the universal applicability of the patent system.

²¹<http://swpat.ffii.org/papers/hbr-thurow97/hbr-thurow97.en.html>

²²<http://swpat.ffii.org/papers/uplr-newell86/uplr-newell86.en.html>

²³<http://www.bustpatents.com/corrupt.htm>

- **Third Paradigm between Patent and Copyright Law**²⁴

A computer program is a linguistic work and a virtual machine at the same time. Neither copyright nor patent law were designed with computer programs in mind. Some scholars and politicians have therefore argued for a “Third Paradigm between patent and copyright law”, also called *pecially tailored right* or *ius sui generis*. Others have called abstract-logical ideas a “nobody’s land of intellectual property” and demanded that it should be kept free of all property claims. Beside the granting of exclusion rights, there are other ways in which information innovation can be stimulated and rewarded.

²⁴<http://swpat.ffi.org/analysis/suigen/swpatbasti.en.html>