

# **DG IV Bakels 2002-06-19: The Patentability of Computer Programs**

<http://swpat.ffii.org/papers/eubsa-swpat0202/dgiv0206/index.en.html>

Workgroup

[swpatag@ffii.org](mailto:swpatag@ffii.org)

2003-09-18

A study on software patentability which was commissioned by the European Parliament's Research Directorate as a reference for deliberations about the European Commission's proposal to make all useful ideas patentable. Unlike most other studies commissioned by EU institutions, this one does not seek to hide the problems with software patents and takes a refreshingly undogmatic look. Bakels finds that the proposed directive fails on all its proposed goals (clarification, harmonisation etc) and contains numerous inconsistencies. Concerning the impact of software patents, Bakels finds that "it is crucial to change the long-standing tradition of patents being granted for relatively simple inventions". Yet Bakels offers no hints as to how this long-standing problem can be solved and does not properly assess how some other possible filters such as concreteness and physical substance (technical character) or "invention character" (laid out in Art 52 EPC) relate to it.

# Contents

<b>1 Strong points</b>	<b>4</b>
<b>2 Weak points</b>	<b>5</b>
<b>3 Key points</b>	<b>8</b>
<b>4 Quotes</b>	<b>10</b>
<b>5 Annotated Links</b>	<b>11</b>

- **Bakels & Hugenholtz 2002-06-19: Study on Software Patentability for European Parliament<sup>1</sup>**

PDF original of the study commissioned by the Research Directorate of the European Parliament, written by Reinier Bakels and Prof. Bernt Hugenholtz from Amsterdam University.

- **Xavier Drudis Ferran comments on the DG IV Bakels & Hugenholtz Study<sup>2</sup>**

Source of much of the fiii page on the same study.

- 

Nice to see that finally official reports recognize problems with software patents, and there are a few good points in the study, although it isn't 100% right. At least, and this is refreshing news, it is readable and plays no word games and sophistry, as was the norm with DG Internal Market sponsored studies.

see

- **IPI 2000: The Economic Impact of Patentability of Computer Programs<sup>3</sup>**
- **PbT Consultants 2001: Summary Report on EU Software Patentability Consultation<sup>4</sup>**

In general one can imagine that the authors see more than they say, but shy away from criticising institutions too much.

Although we'll see some weak points later, the study is a better basis to build on than most other EU documents, and therefore it is very much welcome.

For the more important points see "Key points" below

---

<sup>1</sup><http://www.europarl.eu.int/meetdocs/committees/juri/20020619/SoftwarePatent.pub.pdf>

<sup>2</sup><http://patents.calu.info/dgiv.html>

<sup>3</sup><http://swpat.ffii.org/papers/indprop-ipi00/index.en.html>

<sup>4</sup><http://swpat.ffii.org/papers/eukonsult00/softanalyse/index.en.html>

# 1 Strong points

1. We like the perspective that patent law is economic law, and there is no moral justification for it, nor are patents a natural right of inventors. This is very basic, but is often ignored by the Commission reports, or simply by ordinary people. It is very appropriate to state it as a reminder.
2. The study clearly explains (3.6) that the TRIPs agreement does not impose software patents<sup>1</sup>, and that is good too, since some pretend TRIPs to have already decided our policy for us.
3. It also recognizes (4.1) that prosperity of the software business is not a consequence but a cause of extension of patentability to software. That is true indeed. Microsoft has been cited in other CEC-sponsored papers as having hundreds of patents and having done well, but they never say MS grew without patents and has only started using them strategically when it was large and wanted to keep its monopoly.

In fact in 4.5, when it says:

*This is not the place to speculate about possible strategies of commercial software vendors, but if they would be able to compete using legal means such as the patent system, there is no guarantee they would not do that.*

It misses the recent (barely before the date of the document) news from Microsoft according to which they plan<sup>2</sup> to assert patents against free software compatible with their CIFS specification for sharing files and printers in a network. This is a clear example of their strategic use to keep their monopoly.

It further states (4.3.1) that software innovation exists without the incentive of patents, and therefore software patents are unnecessary and counterproductive.

4. It shows some of the weak points of the COM(2002) 92 directive proposal, such as e.g. the fact that the technical contribution requirement in the directive is meaningless (5.2).
5. It denounces patent inflation (6.2), but unfortunately restricts its outlook to the problem of “trivial patents”. Trivial patents are indeed a serious problem, and should be dealt with before any extension of the patent system (and before the EPO is granted new powers with the Community Patent). But triviality is not the only problem, and the study ignores the evidence that patent inflation is connected not only to a lowering/formalisation of non-obviousness requirements but also to an erosion of the invention/technicality concept and to a series of other factors.
6. It proposes a “patentability observatory” (6.3), i.e. an independent institution to monitor the patent system. This is very similar to the supervisory committee in the EuroLinux demands<sup>3</sup>). This idea should be worked on.
7. We also generally agree on section 6.4. And most of the other points not explicitly commented are useful.

---

<sup>1</sup><http://swpat.ffii.org/analysis/trips/index.en.html>

<sup>2</sup><http://swpat.ffii.org/players/microsoft/index.en.html>

<sup>3</sup><http://swpat.ffii.org/papers/eubsa-swpat0202/demands/index.en.html>

## 2 Weak points

On some issues, the study seems not to be very well informed:

1. The study rightly emphasizes the problems of software patents for SMEs, but fails to note that most software jobs are in SMEs. We do not have the data handy for Europe, but statistics for Catalonia, say<sup>1</sup> that 2/3 of employees in firms with more than 10 workers in information and communication technology work in firms with less than 200 employees. These companies could be seriously damaged if the directive is approved. If the same proportion holds for Europe (as is likely) then saying simply (4.2.2) “The software industry is very diverse. Again, generalisations should be critically examined” is correct, but fails to note that certain generalisations are nonetheless possible and do not speak for the directive proposal.
2. The study downplays the value of economic research work which has so far been done on the subject. Most economics analyses demonstrate that software patents are a bad idea. Of course they can’t be absolutely certain of the outcome of all the possible scenarios, but in a field where direct experimentation is not possible, all we have as a basis for our decisions are these informed and rational speculations. We should assess them critically, but not simply discard them as inconclusive, just because they do not meet unrealistically rigid expectations of exact science.
3. Section 4.1 downplays the meaning of popular opinion and consultations replies.

*Apparent majorities may or may not be meaningful*

it says. In the Commission’s consultation, for example, by reading some hundreds of replies<sup>3</sup>, and then the study of replies to the consultation, it is easy to see that it was not fair<sup>4</sup>. Although the study does not reflect the replies, it shows more than 90% of respondents were against software patents. This is not an “apparent majority”, it is not even simply “widely varying opinions” it is an overwhelming near-consensus. The study might not be a statistical significant sample, but neither was it a sample of only software patents opponents. In fact the organizers were pro software patents, and the consultation was equally open to everyone. It is hardly appropriate in a democracy to downplay the significance of majorities in this way. After all we don’t leave empty seats in Parliament just because abstention rate is higher than we’d like.

4. Section 3.1 fails to note the traditional distinction between copyright and patents. They were not meant to overlap<sup>5</sup>. Certainly you can devise a system like in the US where software can get both patents and copyright, but then software would be the only work covered by both forms of property, and patents being a stronger property would erode the protection offered by copyright. The study recognizes implicitly the potential for those damages (SMEs preferring copyright and being threatened by patents that they could inadvertently infringe and then being unable to exploit their copyrighted original work), but then assumes that both can coexist for the same subject matter.

---

<sup>1</sup>see Statistics on information and communication technology market from the Generalitat de Catalunya<sup>2</sup>

<sup>3</sup><http://petition.eurolinux.org/consultation/index.html>

<sup>4</sup><http://swpat.ffii.org/papers/eukonsult00/index.en.html>

<sup>5</sup>see e.g. BGH 1976-06-22: Dispositionsprogramm<sup>6</sup>

5. Section 4.3.1 is generally shy, in that it shows compelling reasons to believe software patents are more harmful than helpful in promoting innovation, but concludes only that “[the stifling effect on innovation] may be true as well”. Here it also misses one clue: the fact that free software is an increasing and powerful phenomenon shows that not only the software market has enough incentives to innovate without patents, but that it has also enough incentives to disclose inventions in a vastly more useful way for this particular field than patents in inconvenient repositories: with publically available documentation and source code in the internet, that not only explains the invention, but allows anyone to directly test and use it, and any person skilled in the art to modify to adapt it to their needs (not merely reproduce it from scratch like patents do).
6. Section 4.5 shows terminologically that the author is not very familiar with free software, when it opposes “open source” to “commercial software”. You can have any combination of commercial or non-commercial, gratis or priced, “open source” or “closed source” software. Clearly much “open source” software is commercial, and many companies live on it.
7. When section 5.2 comments in the directive, it is a little too soft on the important problems it finds. The directive explanatory memorandum is full of pro software patent propaganda and dogma, with little rational discourse. For a better critique of the directive see CEC & BSA 2002-02-20: proposal to make all useful ideas patentable<sup>7</sup>. The authors seem to have had looked at this critique, but they don’t cite it, only speak of “early reactions to the proposal [...] from the Open Source Software movement and its supporters.” In fact Eurolinux is one of the most active opponents of the directive, and it is an alliance by both “open source supporters” and hundreds of organisations and companies that write and use proprietary software.

In this section 5.2, and specially in section 7:

*Given these uncertainties, it would be more appropriate to concentrate on legal security and harmonisation of law than to contemplate major substantive changes in patent law. This is indeed the route chosen by the European Commission in the proposed Software patent directive, even though the explanatory memorandum accompanying the proposal suggests that software patents are good for the European economy.*

We get the impression the directive proposal has not been fully understood. The directive carefully sets several provisions to avoid any software from not being patentable and forces courts to uphold substantive changes in patent law that have been slipping through currently unenforceable EPO case law. The “technical contribution” requirement is only a dodge, because the directive defines software as being a technical field. Therefore any software contribution is technical and any software is patentable. In fact, when we tried to imagine amendments, we saw that by amending only few articles, the directive would still apparently allow for any software patents.

Any one of the following provisions alone would probably suffice to allow for unlimited patentability:

- Article 2 defines computer programs to be inventions, that would make them patentable subject matter, although art 52 EPC excludes them.
- Article 3 says software belongs to technology. That would make it patentable and would void any technical contribution requirement, since software contributions would be technical contributions and any piece of software represents a software contribution.
- Article 4 completes the empty technical contribution requirement outlined in article 2 and anyway takes care to include non-technical features in the inventive step, so that

---

<sup>7</sup><http://swpat.ffii.org/papers/eubsa-swpat0202/index.en.html>

the only sane delimiting criteria for subject matter cannot be applied<sup>8</sup>. Even if some amendment would leave software as non-technical, innovations in software would still cause inventive step and make programmed conventional computers inventions (with inventive non-technical features).

- Article 5 allows software claims as long as they are dressed up in language involving conventional computers or networks.
  - Article 7 and 8 set up minimal control measures, which are not allocated any budget, to let the patent system continue inflation without proper assessment of its effects on economy and society.
8. The study embraces the CEC/BSA view according to which the understanding of “what is technical” is likely to evolve with the rise of “new technologies”. This is an excuse for leaving free way to the EPO to decide on its own and forces courts to uphold any patents, rather than setting clear limits by law. If the aim is to let the EPO do what it wants, then we should not need a directive.
  9. The fact is that precise definitions of “technical invention” exist<sup>9</sup>, have been used in the past, and are still useful and applicable if one cares to try to tune the patent system for the benefit of society. Occasional technological revolutions will hardly affect philosophical categories such as matter and mind, information and physics, and those can be used to build a persistent framework that is relevant today as it was in the seventies.
  10. Also, there’s somewhere a reference to patents being good for SMEs to raise venture capital. This is (if at all<sup>10</sup>) a temporary effect that will vanish as soon as investors realise that software patents are no good for SMEs and that they are not a stamp of technical excellence, as the EPO itself admits<sup>12</sup>.

---

<sup>8</sup><http://swpat.ffii.org/papers/eubsa-swpat0202/index.en.html>

<sup>9</sup><http://swpat.ffii.org/analysis/invention/index.en.html>

<sup>10</sup>see also *Converting Capital Into Software That Works*. Joel Spolsky<sup>11</sup>

<sup>12</sup>see also *EPO: Patents not a stamp of technical Excellence*.<sup>13</sup>

## 3 Key points

The study fails to see a delimiting criteria for patentable subject matter, and surrenders to its own inability, ignoring traditional distinctions that are well rooted in philosophy, stand the pass of time and simply work.

It focuses the problem on trivial patents. This is a huge problem, but not the only one, and we must not let this one hide other problems. Specially because the triviality problem does not have an easy solution<sup>1</sup>. Or how high should the “inventive height” of a software patent be? 3 Lemelson? 10 Lemelson? How do we define the measuring unit “Lemelson”?

One obvious solution to a large part of the problems lies in combining technicity and novelty in a way, as it used to be the case in the EPO examination guidelines of 1978<sup>2</sup>. Clearly an innovation should not be patentable or unpatentable depending of the drafting of the claim wording. We must look beyond words into meaning and ask whether the application teaches new insights into the use of controllable natural forces for a predictable result. It is not enough that the invention is new and it has physical effects. The physical effects must be new, we should not allow patents on new uses of already existing machines such as computers, whatever electrical currents or effects their peripheral devices have on the physical world. If the innovation lies in logics and calculation to achieve effects through a know machine and known peripherals, using natural forces in known ways, the fact that the effects are new is not sufficient, because we get no new insight in technical matters, just a new application of already available knowledge.

This “technical invention” criterion has its rationale in the rule that any new discovery that “only” needs rational thought would appear sonner or later and does not need monopolies on ideas as an incentive, and risks putting any rational being (specially computer users) under infringement, while those inventions that require empirical observation of the physical world are likely to require experimentation, laboratories and infrastructure, maybe would not be possible without important incentives as patents, and can only be infringed by those manufacturing (or commercially using) patented inventions.

Remember that the scarce human ressource in software (and any information, logic or business innovation) does not consist in the ability to conceive ideas that can be patented, but in writing the code (or building the business, etc.) that puts such ideas to work. This is already protected by copyright (at least for software) and other mechanisms and does not benefit from patents. On the contrary, creativity would suffer from the artificial scarcity of ideas that patents create.

Please see the references for better explanations of the vital concept of the technicity criteria to apply. A proper application of this criterion would not completely solve the triviality problem, but it would eliminate the most objectionable patents, whichconstitue about 3% of the number of patents which the EPO grants every year, and it would do so in a reliable and predictable way.

The study is wrong in identifying triviality as the real cause of concern especially for free/opensource software. Even non-trivial software patents, as rare as they may be, are harmful. In fact algorithms can be classified into an abstract and all-encompassing type and a more concrete and trivial type. The alternative to triviality is abstractness, whose patentability is just as undesirable. The study fails to take into account work that has been done by computer scientists on this “problem of abstraction”, see

- Newell 1986: The models are broken, the models are broken!<sup>3</sup>

---

<sup>1</sup><http://swpat.ffii.org/analysis/trivial/index.en.html>

<sup>2</sup><http://swpat.ffii.org/papers/epo-gl78/index.en.html>

<sup>3</sup><http://swpat.ffii.org/papers/uplr-newell86/index.en.html>

- Tamai 1998: Abstraction orientated property of software and its relation to patentability<sup>4</sup>

A more basic concern about software patents is in their epistemological nature. This is well addressed by the technicity criterion but not by the non-obviousness criterion.

Patents are monopolies in the use of some machine in exchange for publication of information on that machine. When the machine becomes software, and therefore machine=information patents collapse into “monopolies in the use of some information in exchange for publication of information”. So we are allowing people to monopolise information in exchange for publishing information that cannot be used because is monopolised. The fact that publication equals manufacturing in software (and other logical creations) makes patenting software (and all other logical achievements) absurd.

The basic difference that distinguishes software from other fields is that it is information. The study fails to see that, because this line does not only separate software from mechanical engineering, but also excludes other logical achievements from patentability (like VHDL hardware (that is designed by feeding logic programs to a machine that builds the hardware), business methods, social interactions, etc.).

The study keeps reiterating that no criteria can distinguish desirable from undesirable software patents, although such criteria exist and have been successfully used for decades. A *technical invention* is a *teaching of cause-effect relations in the use of controllable physical forces*. This allows for patents on real computer-implemented inventions, such as for example a chemical engineering process that needs certain changes in pressure and temperature controlled by a computer, when there was not previously know that those changes obtained the desired effects. The fact that a program controls the process does not mean that the process is not patentable. We would not be patenting software as such, but a technical invention that happens to use a program. Everybody would be free to use the same program concepts (respecting copyright if any) for other purposes (like a virtual simulation of the process), but they would infringe when performing the chemical process with real machinery and substances. This is the useful meaning of the clause “as such” in EPC art 52, you cannot patent software as such, but can patent inventions that include software when the innovation is not in information but in the material world. Thus, a program alone, on a carrier, in transmission, working on a conventional computer, or otherwise would not infringe. It is not enough to take a conventional computer, argue that it uses electrical currents or known devices in known ways and pretend that that allows patenting any program when the application is properly drafted.

The study has apparently not understood this delimiting criterion for patentability and therefore infers unlimited patentability from a difficulty in defining software.

The basic underlying problem of this study may be that it stops short of asking the responsible governmental institutions to confess that they have committed serious errors. In the USA in 1994, the most well-respected software business pioneers called upon the government to recognize its errors and declare previous decisions of various administrative and judicial organs as illegal<sup>5</sup>. In Europe, the errors are even more evident than in the US. The European Patent Organisation and the european governments who are sitting on its board of directors have acted against the letter and spirit of the law and against the interests of the innovators and the public at large. Face-saving and problem-solving are in conflict. In this context it may be difficult for studies in general and specially those commissioned by state institutions not to opt for some degree of face-saving.

---

<sup>4</sup><http://swpat.ffii.org/papers/ist-tamai98/index.en.html>

<sup>5</sup><http://swpat.ffii.org/archive/quotes/index.en.html#autodesk94>

## 4 Quotes

*In theory non-obviousness is not enough; inventions must meet a certain standard of inventiveness. ...*

*In practice however, a novel invention is rarely denied a patent, unless it is completely obvious to “a person skilled in the art”. To be sure, this is by no means unique to software or business method patents. Unfortunately, courts may not be able to raise this standard, as it is firmly rooted in the tradition of the patent system. A major departure from current practice in this respect would be beyond the authority of the courts. The “patent inflation” caused by granting “trivial” patents on a routine basis in our view calls for a fundamental debate, and eventually legislative action.*

*SMEs apparently do not like patents at all, despite extensive promotion efforts by proponents of the patent system.*

## 5 Annotated Links

- **Breepunt.nl: Bakels: Voorbeeld nieuwsbrief**<sup>1</sup>

Dutch article by Bakels one year later. We have an english<sup>2</sup> and german<sup>3</sup> version also.

- **Reinier Bakels**<sup>4</sup>

Staff home page at Instutut voor Informatierecht (Institute for Information Law). A less complete english version<sup>5</sup> is also available. The dutch page shows pointers to a few more texts which Reinier Bakels wrote on the subject.

---

<sup>1</sup><http://www.breepunt.nl/artikel.asp?Artikel=863&art=Yes>

<sup>2</sup>[bakels-swpat0306.en.pdf](#)

<sup>3</sup>[bakels-swpat0306.de.pdf](#)

<sup>4</sup><http://www.ivir.nl/medewerkers/bakels.html>

<sup>5</sup><http://www.ivir.nl/staff/bakels.html>