

Software patents in Europe: a short overview

FFII Software Patents Workgroup

Contact: plen0309@ffii.org

22 September 2004

Contents

1	What is a software patent?	1
1.1	General	1
1.2	Juridically	1
2	What are software patents used for?	2
2.1	What is software?	2
2.2	Computer-implemented inventions	2
2.3	Software patents	2
3	What is wrong with the Commission and Council proposals?	3
3.1	Summary	3
3.2	Examples of harmful and/or ambiguous articles	3
3.2.1	Article 5.2: program claims	3
3.2.2	Article 4.1: computer programs as such	3
3.2.3	Article 4.2: further technical effects	4
3.2.4	Article 2: pen-and-paper-implemented inventions	4
4	What do we want?	5
4.1	Only Exclusions from Patentability can help “harmonise the status quo”	5
4.2	Freedom of Publication	5
4.3	Freedom to use computers in office and network environments	5
4.4	Freedom of interoperability	6
5	A few studies and opinions regarding software patents	6
5.1	An Empirical Look at Software Patents [BH03]	6
5.2	Innovation in Germany – Windows of opportunity [Hof04]	6
5.3	Opinion of the Economic and Social Committee [Ret02]	7
5.4	The Digital Dilemma: Intellectual Property in the Information Age [NRC00]	7
5.5	To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy [ftc03]	8
5.6	Discussion of European-level legislation in the field of patents for software [BH02]	8
5.7	Opinion Committee for Cultural Affairs and Youth of the European Parliament [Com03a]	8
5.8	Opinion Committee for Industry and Trade of the European Parliament [Com03b]	9

A	History of the directive	10
A.1	The European Patent Convention and the European Patent Office	10
A.2	Gradual introduction of software patents	10
A.3	Rewriting the EPC	10
A.4	Introducing the directive	10
A.5	Opinions	10
A.6	The First Reading in the EP	11
A.7	The Council and the Commission	11
A.8	The Council Working Party	11
A.9	The Council vote of 18 May 2004	11
A.10	What's next?	11

1 What is a software patent?

1.1 General

Simply put, software patents are patents that cover computer programs. They allow the owner to forbid other people from using or commercialising such computer programs. Some examples of already granted, but as of yet largely unenforceable, European software patents:

- EP807891: a patent on the electronic shopping cart. This patent covers all possible computer programs that contain functionality which allows a user to select several items to buy while browsing through an online shop, instead of forcing him to pay every time he has selected some items on a particular web page. This patent was granted to the US company Sun Microsystems.
- EP933892: a patent on distributing video data over a network for playback in real-time at the user's end. This patent covers all possible computer programs which can download video data and play back this video data at the request of the user. This patent was granted to the US company Greenwich Tech.
- EP986016: a patent on providing consumers with commercial information if they request so. This patent covers all computer programs which are connected to databases with commercial offers from retailers and manufacturers, and which send these offers to consumers if they requests so. One can thus avoid this patent by spamming the offers instead. This patent was granted to the US company Catalina Marketing Int.

These are just a few examples of the more than 30,000 already granted European software patents against the letter and the spirit of the law. Twenty more can be found on our illustrated patented webshop at <http://webshop.ffii.org>.

1.2 Juridically

Juridically, software patents can be obtained by writing a patent in one of two ways:

- Using *process claims*. In 1986 the European Patent Office (EPO) started granting patents which covered computer programs, but which were presented in the guise of process claims (which are normally used to patent e.g. a new way to conduct a chemical reaction; i.e., when one wants to patent the way of doing something). They are typically phrased as follows:

1. Process for [using a computer, keyboard, screen, hard disk, ...], characterised by [whatever the class of computer programs one wants to patent does]

The patents granted on this basis are considered to be hypothetical, because the program itself, when distributed on a disk or via the Internet, does not constitute a process and is thus not covered by such patents. Only actually running and using them is an infringing activity in this case.

The end result is however still that more or less all practical uses of a computer program (namely running it on a computer) are covered by such a patent, despite the fact that computer programs are excluded from patentability by law.

- Using *program claims*. Such claims were allowed by the EPO starting from 1998 and are typically phrased as follows (example from the granted European software patent EP1073245 on a method to evaluate visits to web pages):

17. Computer programme for implementing a method as claimed in any one of claims 2 to 12 [...]

18. Computer programme as claimed in claim 17, recorded on a medium readable by a computer.

In this case, all computer programs that perform something described in the patent claims are monopolised. Since such claims only cover computer programs, they really are claims on “computer programs as such”, in literal contradiction with the law.

It is as if instead of monopolising a chemical reaction, one would monopolise “a book, electronic document or other information entity describing how to perform the chemical reaction from claim 1”. Consequently not only the use, but also the publication and distribution of such programs can be prohibited by the patent owner (just like in case of the chemical reaction, one would be able to not only prohibit others from performing that chemical reaction, but also from even spreading descriptions of how it works).

2 What are software patents used for?

2.1 What is software?

Software is simply a description of something (it is just text in some language after all) and software patents allow one to monopolise whatever can be described in software (e.g., selling things over the Internet – EP0803105). The actual implementation/description is already protected by copyright, and that’s how businesses and individuals alike have been successfully protecting their investments in software development until now.

2.2 Computer-implemented inventions

One does not need patents on “computer-implemented inventions” to keep real inventions patentable when they are described using software. A computer program is just an instruction manual in a language that can be interpreted by a machine. A computer program executed by a computer that steers lab equipment to perform a chemical reaction, is functionally identical to a technical manual describing the same thing.

Even though the chemical reaction could be patentable, the technical manual can’t be and publishing it cannot constitute a patent infringement. Additionally, the fact that the chemical reaction is described using words in a book does not make the reaction itself unpatentable.

Consequently, similar rules were made for patentability regarding software in the European Patent Convention (EPC) of 1973: achievements described using software may be patentable, but the fact that software is used to describe them does not have any influence on their patentability, and the software itself is not patentable either.

2.3 Software patents

One only needs software patents (or patents on “computer-implemented inventions” as defined by the Commission proposal) in order to be able to patent achievements which are currently

unpatentable, such as mathematical algorithms and business methods. Along with computer programs, these are all achievements which are currently excluded from patentability by the EPC. Software patents are used to get around these exclusions, in direct contradiction with the law.

3 What is wrong with the Commission and Council proposals?

3.1 Summary

Software patents as introduced by the Commission allow one to get around the EPC by claiming that “a computer program executed by a computer” is not the same as “a computer program as such”. The end result is that one can patent all useful applications of a computer program (running it on a computer) in the form of a process claim.

The Council version goes even further and implies that the exclusion of “computer programs as such” from patentability only means that one should not be able to patent the source code of individual computer programs. This interpretation does not make sense, because the protection one would obtain by doing so would be less than that offered by copyright, and additionally would cost money (as opposed to copyright protection, which happens automatically and is free). Nevertheless, they used this argument to justify the introduction of program claims in their version of the directive, and thus went even further than the Commission.

3.2 Examples of harmful and/or ambiguous articles

All articles mentioned below refer to the Council version of 18 May 2004 (available at <http://tinyurl.com/3ocdx>), but the last two are semantically identical in the Commission version.

3.2.1 Article 5.2: program claims

A claim to a computer program, either on its own or on a carrier, shall not be allowed unless that program would, when loaded and executed in a computer, programmed computer network or other programmable apparatus, put into force a product or process claimed in the same patent application in accordance with paragraph 1.

This article introduces program claims. Even if the rest of the directive excluded software patents, this one paragraph would overturn it all. The reason is that it allows patents on computer programs “on their own”, as long as they describe something which is claimed elsewhere in the patent.

Notice the double negation, and the fact that the “unless” condition can always be fulfilled by including an appropriate process claim.

3.2.2 Article 4.1: computer programs as such

1. *A computer program as such cannot constitute a patentable invention.*
2. *... inventions involving computer programs, whether expressed as source code, as object code or in any other form, ...*

Together these changes, introduced by the Commission during the Council session of 18 May 2004, imply that the exclusion of “a computer program as such” from patentability in the EPC only applies to the source code or object code of individual computer programs, instead of to computer programs in general.

This would mean that the intention of the writers of the EPC was that e.g. Microsoft Word XP should not be patentable, but that one should be able to patent the principle of “word processors” (as long as that is new and non-obvious at the time this patent is applied for).

3.2.3 Article 4.2: further technical effects

A computer-implemented invention shall not be regarded as making a technical contribution merely because it involves the use of a computer, network or other programmable apparatus. Accordingly, inventions involving computer programs, whether expressed as source code, as object code or in any other form, which implement business, mathematical or other methods and do not produce any technical effects beyond the normal physical interactions between a program and the computer, network or other programmable apparatus in which it is run shall not be patentable.

This article codifies the “further technical effect doctrine” of the EPO, which the EPO itself has admitted to be confusing, undesirable and only a stopgap until patents on software as such are legalised ¹.

3.2.4 Article 2: pen-and-paper-implemented inventions

(a) “computer-implemented invention” means any invention the performance of which involves the use of a computer, computer network or other programmable apparatus, the invention having one or more features which are realised wholly or partly by means of a computer program or computer programs;

(b) “technical contribution” means a contribution to the state of the art in a field of technology which is new and not obvious to a person skilled in the art. The technical contribution shall be assessed by consideration of the difference between the state of the art and the scope of the patent claim considered as a whole, which must comprise technical features, irrespective of whether or not these are accompanied by non-technical features.

The problems with this article are manifold:

- “A computer program executed by a computer” fulfills the definition of “computer-implemented invention”.
- The definition of “technical contribution” is worded ambiguously and can be interpreted as meaning that the technical contribution can consist entirely of non-technical features (and that only “the scope of the patent claim as a whole” must contain technical features, which can be done by simply mentioning the use of a computer somewhere).
- The word “technical” is not defined anywhere, yet all defined boundaries of patentability hinge on the meaning of this term. Pretty much everything one does on a computer has “technical character” according to the EPO, and there is nothing in these definitions (nor in article 4) that contradicts this interpretation.

¹See Note 1 on page 5 of <http://www.european-patent-office.org/tws/appendix6.pdf>

Recently, the Board of Appeals of the EPO continued this line of reasoning and stated ²

The Board is aware that its comparatively broad interpretation of the term "invention" in Article 52(1) EPC will include activities which are so familiar that their technical character tends to be overlooked, such as the act of writing using pen and paper.

The Council proposal implies that this kind of "logic" is a valid reinterpretation of the law. How long will it still take before we can expect another directive proposal, this time regarding the patentability of "pen-and-paper-implemented inventions" in order to "codify the status quo" and "to prevent an even further drift towards US-style patenting"?

4 What do we want?

Some core points and articles can be found below. More information on this subject is available at <http://swpat.ffii.org/analysis/needs/>. The version of the directive approved by the European Parliament in first reading can be found at <http://tinyurl.com/3g4yh>.

4.1 Only Exclusions from Patentability can help "harmonise the status quo"

We can accept almost any directive, as long as it consists only of clear and simple exclusions from patentability. Art 52(2) EPC consists of such exclusions. It says in clear and simple terms what is not an invention in the sense of patent law (e.g., computer programs).

We cannot accept inclusive language, such as *X shall be patentable* where X is not in turn delimited in clear and simple exclusions. Council-style language of the sort *X shall not be patentable, unless [condition which is always true]* is even less acceptable.

Article 2 (all paragraphs) of the European Parliament takes care of this.

4.2 Freedom of Publication

Our constituents' basic interest is to keep the software free from patents, regulated by copyright only. I.e., even if there are patents on the much cited "anti-lock braking system (ABS)", "washing machine", "intelligent vacuum cleaner" etc, they must apply only to the makers and users of such devices and not to people who create or provide software (= control logic, similar to user manuals) for these devices.

The European Parliament has fulfilled this demand with Article 7 (paragraphs 1 till 3).

4.3 Freedom to use computers in office and network environments

We can not accept restrictions on the use of equipment that consists of general-purpose computers only. The Parliament has solved this problem by a clear and simple exclusion statement in the spirit of Art 52 EPC, by stating that data processing does not belong to a field of technology (Article 3). At the same time, this article guarantees TRIPs compliance.

²See page 16 of <http://legal.european-patent-office.org/dg3/pdf/t030258ex1.pdf>

4.4 Freedom of interoperability

In their Articles 6, the Commission and Council guarantee the right to reverse engineer patented techniques (which cannot be forbidden using a patent in the first place), but no right to use the knowledge gained from this process (which is what the patent does forbid) for any purpose whatsoever.

Interoperability –the ability to exchange information with other programs and systems– is however extremely important in the computer world. Just look at the Internet, which consists of a very large number of different kinds of systems that all have to be able to talk to each other.

Consequently, enabling the usage of patents to forbid conversion between communication standards is very harmful in such an environment. Article 9 of the European Parliament takes care of this problem, although its wording could be changed so it only applies to interoperability instead of to all “significant purposes” (as suggested in the Council by Luxembourg).

5 A few studies and opinions regarding software patents

Below is a short overview of studies and opinions regarding software patents. A more extensive list can be found at <http://swpat.ffii.org/xatra/cons0406/par1/>.

5.1 An Empirical Look at Software Patents [BH03]

Empirical study on the effects of software patents on investments in innovation in the US.

- Software patents have in the US resulted in a transfer of resources from R&D to patenting activities.
- More patents meant less innovation, even within the companies that patented most.
- Most software patents are owned by large hardware companies and obtained for strategic purposes rather than for preventing imitation of products.
- Software patents hinder instead of encourage innovation in fields where most innovation is incremental, such as in software development.

5.2 Innovation in Germany – Windows of opportunity [Hof04]

Report of Deutsche Bank for the German government regarding how to promote innovation.

- “Stronger IP protection is not always better. Chances are that patents on software, common practice in the US and on the brink of being legalised in Europe, in fact stifle innovation. Europe could still alter course.”
- “Opportunity 3: Set up a well-balanced IP protection regime that keeps fostering the creation and diffusion of ideas.”
- “Measures to take. The German government is among the tentative critics of the EU software patent bill. This position should be bolstered, by (1) putting forward academic evidence and (2) making SMEs’ concerns heard. SMEs are crucial providers of pathbreaking innovations, but would be most adversely affected by patentability.”

5.3 Opinion of the Economic and Social Committee [Ret02]

ESC is the main consultative organ of the EU. Its opinion was approved by the EP's plenary vote in first reading.

- Commission text allows patents on software executed by a computer.
- Commission text simply codifies legally questionable EPO practice.
- Commission text does not prevent patents on business (or on any other) methods.
- Doubts about intention of Commission, which talks about several irrelevant things (such as piracy) in its introduction.
- “No effective economic analysis has shown the alleged benefit for SMEs-SMIs of patents for ‘computer-implemented inventions’.”
- “It is hardly plausible to have us believe that the directive would only be a sort of reversible three-year experiment, at the end of which an assessment would be made.”
- “For the most part, the opinion that has been credited is that of a dozen large software houses, most of which are not European. Furthermore, an opposing opinion from other large firms has been ignored, as have some counter-proposals which advocate a sui generis regime or an adapted utility model.”
- “Is it wise in today's world to widen the scope of patents, tools of the industrial age, to intellectual works which are immaterial, such as software, and to the results of running software on a computer?”

5.4 The Digital Dilemma: Intellectual Property in the Information Age [NRC00]

A book from the US National Research Council.

- Granting of software patents started in the US without oversight from the legislative branch (just like in Europe).
- Doubts about ability of US Patent Office to handle software patent related decisions, and whether it has enough knowledge and prior art information available.
- Software market is different from traditional industries: small or no market in ‘components’, most people write programs from scratch, no consultation of patent literature, high chances of infringement.
- Innovation in software development occurs more rapidly than in other industries, patents often granted after the technology has become obsolete.
- Software patents may cause the software industry to cease being a creative cottage industry, restricting it to large companies that cross-license.

5.5 To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy [ftc03]

The US Federal Trade Commission (FTC) conducted hearings to find out how the patent system promotes and/or inhibits competition in different fields. Its conclusions were bundled in a report with the above title.

- All industries are not the same, in the computer hardware and software industries patents are used more and more for defensive purposes. This results in patent thickets: overlapping and entangled patent rights of different companies, which means one has to obtain a license to all such patents before the product can be commercialised.
- The software industry is characterised by cumulative innovation, low capital costs, rapid consequential innovation and a short life span of products and alternative incentives for innovation such as copyright and Open Source. This is quite different from the hardware industry, biotech and pharmaceuticals.
- Innovation in the software industry is driven by competition.
- Software patents can inhibit consequential innovation and increase the entry cost. Avoiding infringement is expensive and uncertain.
- There are also large problems due to trivial patents.

5.6 Discussion of European-level legislation in the field of patents for software [BH02]

Study ordered by the JURI Committee of the European Parliament, edited by the Directorate General Research of the European Commission.

- Notes general problems with patent system as a whole.
- Problem of ‘trivial patents’ can not be solved by improving examination.
- Software patents have caused a lot of problems in the US (both economical and administrative).
- Requirement of ‘technical contribution’ is too vague in Commission proposal and can easily be circumvented, may even not be relevant by Commission’s own admission (in that it cannot prevent all business methods from being patented).

5.7 Opinion Committee for Cultural Affairs and Youth of the European Parliament [Com03a]

- “Technical” means “application of natural forces to control physical effects beyond the digital representation of information” (Article 2)
- Data processing is not a field of technology (Article 3)

5.8 Opinion Committee for Industry and Trade of the European Parliament [Com03b]

- Publication can never be an infringement (Article 5)
- Interoperability can never constitute patent infringement (Article 6a)

A History of the directive

A.1 The European Patent Convention and the European Patent Office

The European Patent Convention was ratified in a diplomatic intergovernmental conference in 1973, and was the result of harmonising the patent laws of several EU and non-EU countries. This EPC was in turn again incorporated in the patent laws of the various subscribing countries.

Apart from laying down the conditions for patentability (including the non-patentability of computer programs, mathematical methods, rules for performing mental acts, ...), the EPC also provided for the foundation of the European Patent Office.

This institution lies entirely outside the EU, is self-sufficient (it finances itself with the money it receives from granting patents) and is lead by delegates from the national patent offices of the subscribing countries.

As such, an EU directive will never directly affect the EPO, but it will affect the enforceability of the patents it grants. The reason is that these must be enforced in national courts, which fall under national laws that are in turn bound by European directives.

A.2 Gradual introduction of software patents

As described in section 1.2, as of 1985 software patents started to be allowed by the EPO in the form of process claims. From 1998 on, program claims were found to be acceptable as well.

A.3 Rewriting the EPC

In August 2000, a new diplomatic conference was organised with the aim to scrap the exclusion of software from patentability from the EPC. This attempt failed however due to unexpected public resistance.

A.4 Introducing the directive

In 2002, the European Commission then came out with its proposal for “a directive on the patentability of computer-implemented inventions”. Its text completely codified EPO’s practice of granting software patents, except that it did not allow program claims.

This directive is handled under the co-decision procedure and has the reference `COD/2002/0047`.

A.5 Opinions

As shown in the overview of studies and opinions, the European Economic and Social Committee (ESC) literally blasted the Commission proposal in September 2002. The ITRE and CULT committees of the European Parliament also substantially amended the Proposal in their advice to the JURI Committee early 2003, which was responsible for this directive in the EP.

JURI however almost completely ignored all these opinions when making its final version in June 2003. It did retain the guarantee for interoperability, but threw away everything else and on top of that introduced program claims.

A.6 The First Reading in the EP

In its plenary session of 24 September 2003, the European Parliament approved the opinions of ESC, CULT and ITRE. The core articles (2, 3, 5 and 6, afterwards relabeled as 2, 3, 7 and 6) were all amended in a consistent way to this effect.

A.7 The Council and the Commission

Next, it was the turn of the EU Council of ministers to voice its opinion. The amended text was sent to them, along with a note from the Commission stating which amendments were acceptable to them³. More or less only approved JURI amendments were found to be acceptable; the core achievements of the EP were all labeled as “unacceptable”.

A.8 The Council Working Party

Next, the Council Working Party on Intellectual Property (Patents) started to produce a text for the Council to vote on. This Working Party mainly consists of the same delegates from national patent offices that lead the EPO.

They wrote the most extreme text in favor of unlimited software patents seen until now, including a provision for allowing program claims. It was even based on the JURI-proposal and not on the plenary version, as shown by the fact that they note that they delete article 8(d)⁴. The JURI text is the only one which contains a paragraph with this number.

A.9 The Council vote of 18 May 2004

Of course, ministers do not take all decisions on their own, they are advised by knowledgeable people from their administration. In case of the software patent directive, these people were generally the same people that started the practice of allowing software patents at the EPO, who will have to work by the rules laid down by this new directive and who wrote the text on which the ministers had to vote: once again the administrators from the national patent offices.

As a result, the Council reached an (informal) “political agreement on a Common Position of the Council” based on the text produced by their working party.

A.10 What’s next?

Several irregularities happened at the Council vote of 18 May 2004⁵. Normally, turning a political agreement into a Common Position is merely a formality. This was originally planned for 24 September 2004, but now has been pushed back to November 2004.

Many indications point to the item being back at CoRePer for further negotiations, but the delay is officially explained as a consequence of some final juridical-linguistic checking. After the Council approves of a Common Position, the directive will come back to the European Parliament for a second reading.

³<http://register.consilium.eu.int/pdf/en/03/st13/st13955.en03.pdf>

⁴See Article 8(d) in <http://register.consilium.eu.int/pdf/en/04/st09/st09713.en04.pdf>

⁵For details, see <http://swpat.ffii.org/xatra/cons0406/repr/>

References

- [BH02] Reinier Bakels and P. Bernt Hugenholtz. The patentability of computer programmes – Discussion of European-level legislation in the field of patents for software. Study Commissioned by JURI, University of Amsterdam (IViR), April 2002. <http://www.europarl.eu.int/hearings/20021107/juri/study%5Fen.pdf>.
- [BH03] James Bessen and Robert M. Hunt. An Empirical Look At Software Patents. Working paper, March 2003. <http://www.researchoninnovation.org/online.htm#sw>.
- [Com03a] CULT Committee. Opinion of the Committee on Culture, Youth, Education, the Media and Sport for the Committee on Legal Affairs and the Internal Market on the proposal for a directive of the European Parliament and of the Council on the patentability of computer-implemented inventions (COM(2002) 92 - C5-0082/2002 - 2002/0047(COD)). Advice, January 2003. <http://www.europarl.eu.int/meetdocs/committees/juri/20030324/487019en.pdf>.
- [Com03b] ITRE Committee. Opinion of the Committee on Industry, External Trade, Research and Energy for the Committee on Legal Affairs and the Internal Market on the proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions (COM(2002) 92 - C5-0082/2002 - 2002/0047(COD)). Advice, February 2003. <http://www.europarl.eu.int/meetdocs/committees/juri/20030324/490455en.pdf>.
- [ftc03] To Promote innovation: The Proper Balance of Competition and Patent Law and Policy. Report, Federal Trade Commission, October 2003. <http://www.ftc.gov/os/2003/10/innovationrpt.pdf>.
- [Hof04] Jan Hoffmann. Innovation in Germany – Windows of opportunity. Report, Deutsche Bank Research, June 2004. <http://www.dbresearch.com/PROD/DBR%5FINTERNET%5FEN-PROD/PROD0000000000175949.pdf>.
- [NRC00] National Research Council. *The Digital Dilemma: Intellectual Property in the Information Age*. National Academy Press, 2000. <http://books.nap.edu/html/digital%5Fdilemma/>.
- [Ret02] Daniel Retureau. Opinion of the Economic and Social Committee on the “Proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions”. Opinion, European Economic and Social Committee, September 2002. <http://eescopinions.esc.eu.int/EESCopinionDocument.aspx?identifiant=ces%5Cint%5Cint145%5Cces1031-2002%5Fac.doc&language=EN>.