A short overview of the Council and EP texts

FFII Software Patents Workgroup
Contact: plen0309@ffii.org

22 September 2004

Contents

1 What is wrong with the Commission and Council proposals? 1
  1.1 Article 2: pen-and-paper-implemented inventions .................. 1
  1.2 Article 4.1: computer programs as such .......................... 2
  1.3 Article 4.2: further technical effects ............................. 2
  1.4 Article 5.2: program claims ....................................... 3

2 What do we want? 3
  2.1 Only Exclusions from Patentability can help “harmonise the status quo” 3
  2.2 Freedom of Publication ............................................ 4
  2.3 Freedom to use computers in office and network environments ........ 4
  2.4 Freedom of interoperability ....................................... 4
1 What is wrong with the Commission and Council proposals?

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.

1.1 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 run on 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 term “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 European Patent Office (EPO), and there is nothing in these definitions (nor in article 4) that contradicts this interpretation.

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

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”?  

1.2 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, . . .

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 European Patent Convention (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 (which no one never would want to patent anyway, because the resulting monopoly would be both more costly and narrower than the automatic and free protection offered by copyright), 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).

This is in stark contrast with the actual intention we want to preserve, which was that “if the contribution to the known art resides solely in a computer program then the subject matter is not patentable” (as it was literally in the European Patent Office’s own guidelines, as they were into force until 1985).

1.3 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 has been noted as confusing, undesirable and only a stopgap until patents on software as such are legalised.  

No one specialised in computer science or patent law can clearly define what a “further technical effect beyond the normal physical interaction between a program and a computer” is, if only because a program is not physical and thus per definition cannot interact physically with a computer.

Consequently, explanations of this “further technical effect” are based on intuition or examples, and (maybe surprisingly) that’s also how the EPO assesses it. It is also this intuition which results in many granted software patents.

For example, IBM recently grabbed headlines with its announcement that it made 500 of its software patents available to the open source community. We searched for the European versions of those patents in our (still incomplete) database of European software patents, and found 44 granted, 43 rejected and 11 pending patents.

Even if we would conclude from this very superficial analysis that the EPO practice manages to exclude half of the software patents that are granted in the US (although several were rejected because prior art was found, and not because they did not “make a technical contribution”), it still does grant the other half.

\textsuperscript{2}See Note 1 on page 5 of \texttt{http://www.european-patent-office.org/tws/appendix6.pdf}

\textsuperscript{3}\texttt{http://kwiki.ffii.org/IbmEp0501En}
1.4 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.

The program claim will in practice be used for enlarging the scope of what normal patent law might treat under “contributory infringement”.

Without the program claim, it would be up to patent courts to decide when the act of supplying a disk with a ready-made program for performing the claimed processes could be considered to be an act of facilitating patent infringement. The court might decide that the supplying of a disk is to be treated just like the supplying of an operation manual: a legitimate act in all cases. Or they might decide that, unlike the manual, the disk with a customised binary image on it has no other function than to assist in infringing the patent and is therefore not allowed.

With the program claim, the court would still have to engage in the same kinds of deliberations. But it would be less likely that the court would decide that supplying a disk is legitimate in any case.

By giving the patentee an undefined entitlement, which looks like a normal patent claim but in fact fails to describe what the patentee invented and thereby violates the grammar of patent claims, the patent office would be prejudicing or circumventing decisions that normally need to be made at a different level, namely that of determining the forms of infringement (e.g. Par 11 of German Patent Act, a set of questions that does not belong to substantive patent law and is outside of the scope of the European Patent Convention).

2 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/](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](http://tinyurl.com/3g4yh)

2.1 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.
2.2 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 (keeping in mind what we said in section 1.4, in that courts can still overrule this general principle on a case by case basis).

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

2.3 Freedom to use computers in office and network environments

We cannot 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.

This does not exclude devices which perform datatransfer from patentability, nor does it exclude everything which requires datatransfer for implementation purposes. For example, improving combustion in an engine by using better timings for fuel injection will remain patentable under the European Parliament’s version, even if this is in practice implemented using a computer program.

What is excluded are improvements which only lie in datatransfer. Such improvements will also include things which have been patented by companies such as Nokia (e.g. EP0776141, which is a patent on using pin codes to protect a mobile phone), but that’s because they are no different from improvements in general computer programs (in the previous case: using a password to log in on a computer). Economic evidence shows that such patents do more harm than good in so far innovation and competition is concerned 4.

We are moving from an industrial economy to an information economy. That does not mean that the protection means from the industrial era should per definition apply to the innovations from the information society.

Software is relatively “easy” and “cheap” to develop (electronics companies are not moving more innovation from hardware to software because it’s more difficult or more expensive). Therefore, as humans we take up the challenge and combine more (simple) things, resulting in systems which in the end are just as expensive and complex as the old the ones, but which consist of many more individual ideas and innovations, all of which are potentially patented.

After 50 patent licenses in exchange for 2% of the revenue on sales of a product, license number 51 is no longer affordable. Patent pools alleviate this problem somewhat, but still result in large hurdles for small and medium-sized enterprises and put them at the mercy of patent holders. That is why European-wide SME associations such as UEAPME 5, CEA-PME, CEDI and ESBA 6 have spoken out against against software patents.

2.4 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

4 [http://tinyurl.com/65duf](http://tinyurl.com/65duf)
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. Without interoperability, we would have no email and no world wide web.

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).