

## Analysis of the political agreement in the European Council on software patentability (proposed 20002/47/COD directive on the patentability of computer-implemented invention)

### Context:

This analysis has been produced on the basis of what is known of the political agreement adopted by a qualified majority in the Competitiveness Council of 18 May 2004. One will find at the end of this text a detailed analysis of the differences between this agreement and the text tabled by the Irish Presidency in the Committee of Permanent Representatives of 5 May 2004. These differences can be summarised as follows: more smokescreens to hide the fact the text opens wide the door to patentability of underlying ideas of software and information processing methods, and to justify the reversal of those countries that previously committed to not support the Irish text and finally voted for it, notably Germany. The countries that did not vote for the text are Spain (no vote), Belgium, Italy, Austria, and Poland. France, Slovenia and Hungary voted for the text, while insisting on the need to clarify it before it is turned into a formal common position concluding the first reading of the text by the Council.

### Analysis:

The proposed text is in my opinion much worse than the original proposal from the Commission on two counts:

- It has reached further levels of hypocrisy and abstraction. Some recitals are worded in such a way that they seem to indicate a wish to restrict the field of patentability, while all the legally binding provisions in the articles entirely endorse the recent practice of the EPO, and fail to install any solid restriction against monopolising algorithms and information processing methods. Some provisions are achieving new summits of legalese Newspeak<sup>1</sup>.
- The proposal introduces software claims (article 5b) which is considered by all serious analysts as equivalent with the patentability of software as such. This was explicitly recognised in the Commission proposal in a footnote in the explanatory memorandum. I have personally heard repeatedly high-level representatives of EPO stating that the reason for accepting software claims was to make possible for patent holders to go after writers, copiers, distributors and publishers of software.

The 2 points above are combined when one considers on one side the new article 4a that states:

*(4anew) Accordingly a computer program as such, in particular the expression of a computer program in source code or object code or in any other form, cannot constitute a patentable invention*

and on the other side article 5b that states:

---

<sup>1</sup>A fictitious (or was it ?) language introduced by Orwell in 1984, that aims at making it impossible to express uncompliant ideas by emptying words from their meaning.

*Art. (5b) A claim to a computer program, either on its own or in 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.*

Leaving aside the misleading negation in the first sentence of art. 5b, one can summarise the situation that would result from this text by: software, algorithms and data processing methods will be said to be unpatentable, however we will patent them, and give big weapons for those who hold the patents to go after developers and distributors.

Unfortunately, this is not by far the only provision aiming at officialising the plundering of the public domain of ideas and information in the text. The proposal has re-introduced (by ignoring 43 amendments from the Parliament) the worse provisions of the original Commission proposal :

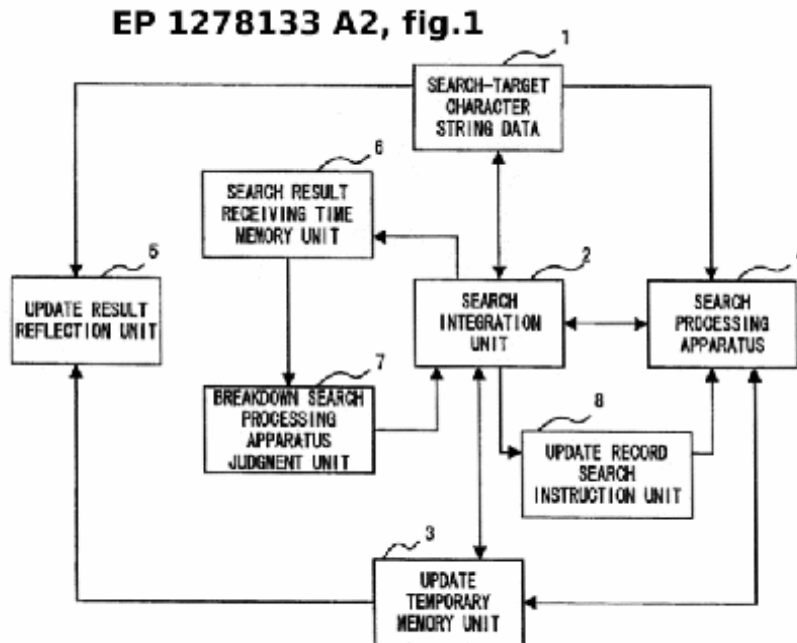
- whereas 6 affirms that computer implemented inventions belong to a field of technology, and thus fall under TRIPS art. 27.1 and other TRIPS provisions on enjoyment of patents. Since a computer-implemented invention is defined at article 2a as anything the performance of which involves a computer, ... such as running software on a computer, it means that any software belongs to a field of technology (in contradiction to the reference to EPC 52.2 in whereas 7). This is an extremely dangerous statement that promotes an interpretation of TRIPS (everything is technology as long as one can find a trick to call it invention) that will have far-reaching consequences, well beyond the field of software. The complex legalese in whereas 12 aims at pretending that the notion of technical contribution can constitute a barrier. However, even recently on 14 April 2004, David Sant, European Liaison of EPO presented in the European Parliament the official interpretation of "technical contribution" that is so wide that it applies to anything when one can say that "technical considerations" have to be taken before starting programming, a situation to which I have never seen an exception in 25 years of software practice and research.

- Article 2b allows to argue of a difference between the patent claim "considered as a whole" (including non-technical features) and the state-of-the-art to decide that there is a technical contribution. The limitation that art. 4a pretends to establish by rejecting patentability of methods that would be claimed to be technical simply because they use a computer while they *do not produce any technical effects beyond the normal physical interactions between a program and the computer, network or other programmable apparatus* has no effect. It is already there in the examination guidelines of EPO, and this does not stop them (nor the patent applicants) from finding ever new ways of claiming that a program has effects beyond this normal physical interactions for reasons such as the fact that it presents information quicker or better, runs faster, etc, that is does exactly what all good software is about. The alinea introduced at article 4a stating that a program as such is not a patentable invention scores high in hypocrisy scales: the program as such may be said unpatentable, but its underlying ideas are definitely made patentable, and the program can be claimed, which means writers and publishers of software can be accused of direct infringement.

- Article 6 is supposed to protect interoperability, but is totally ineffective. It only allows the reverse engineering and decompilation exceptions planned for in the 1991 software protection by copyright directive. But while these exceptions were sufficient in a copyright context, in a patent context, it would remain impossible to implement interoperability even when one knows how to do it, since it would still be infringing on patent(s) in the absence of an explicit exception to patent rights (authorised in article 30 of TRIPS). A new (whereas) states (exact wording

unknown) that companies in dominant position may be asked to share information needed for interoperability, but without creating a compulsory, royalty-free, non-discriminatory license. This will be not only ineffective, but also deeply detrimental to free software, as it will make possible for the holders of patents on interoperability software (made valid by the rest of the text) to require payment of royalties, which is incompatible with the principles of free software.

Only one of the 4 essential points (software claims, putting all software under TRIPS realm, allowing everything to be technical and failing to protect interoperability) would be more than enough to reject the full text.



*Illustration 1: An example of EPO patent illustration or "how to give a technical contribution to a basic algorithm, here a full text search"*

## Changes introduced at the Competitiveness Council of 18 May 2004

Analysis based on incomplete information, as no resulting text has been published. However most of the contents are known.

- (whereas 13) has been deleted. It made very clear that algorithms were made patentable. Unfortunately, now one does not say it clear, but algorithms are still made patentable as "technical solutions in the field of information processing", since the clear exclusion of algorithms and information processing in amendments 69 and 55 from the Parliament, that Germany had proposed earlier to reintroduce in article 2b is not introduced. In summary: more confusion in the text, and as much harm from it.

- The word "new" has been added in the definition of technical contribution in article 2b. That does change a thing, since one can still argue of novelty into the full scope of the claim (including non-technical features). The Council refused again to impose for novelty to reside in the technical characteristics per se, and to identify these characteristics by the fact that physical means are used to obtain physical effects. If one wanted to patent washing machines and braking systems, why would it be a problem? However, the pro-patent lobby wants to patent the processing, presentation and exchange of information.

- The former (whereas 7b) has been turned into a new alinea of article 4a. It states that a

program as such is not a patentable invention, whether in source, binary or other form. This is hardly a surprise since saying otherwise would be saying explicitly that the Council ignores the Munich Patent Convention, and thus clearly illegal. However, the result is the same: since a piece of software becomes a technical solution as soon as one needs to consider technical parameters before starting to program it (case law of EPO).

- Finally a whereas was introduced to state that companies holding dominant positions may be asked to provide information needed for interoperability, but without saying how this will be decided (one would need it to be stated in an article, not a whereas) nor that it should be done with a license that is royalty-free, and non-discriminatory (for instance towards licenses such as the GPL).