

European Commission
DG Internal Market (MARKT/E/2)

Below I have enclosed a proposal for regulation of software patenting. This is just a suggestion for the lines along which I think it is necessary to develop a solid regulation of software patenting. The contemporary regulation are clearly insufficient.

Yours Sincerely

Flemming Bjerke
Centre for Research on
Public Organisation and Management (COS)
Copenhagen Business School
Blaagaardsgade 23C
2200 Koebenhavn N
Phone: (+45) 38152188
Secretary: (+45) 38153815
E-mail: fb.cos@cbs.dk
Private Phone: (+45) 46409102
Private e-mail: Flemming@Bjerke.dk

Proposal for regulation of software patenting

1. Software as such is not patentable.
2. To the extent that a software algorithm is combined with specific machinery or instruments having specific chemical and physical effects that are not related to the storing and processing of the software, the specific combination of software and instrument is not software as such and is patentable.
3. If a specific algorithm has been patented as in 2, it cannot be patented in any other context.
4. If the chemical and physical effects are not physically and chemically new, but practically, symbolically or sensationally new, 2 does not apply.
5. Software consists of one or more programs. A program is a sequence of instructions describing how to perform in a computer certain tasks. An algorithm is the principle or the idea on which the program is build.

Argument

It has been thoroughly argued that patenting of software as such is harmful for the free development of software. Software patenting will reduce competition in the software development industry so that it will be controlled by a small number of globally dominating multinational companies (but perhaps often performed by smaller companies in agreement with the dominating companies). Since the software development particularly takes place in USA and only marginally in the third world, software patenting will consolidate the existing economic world order, in which USA is globally dominating and the third world dominated. Europe will be strongly dependent on USA and not gain its own economic momentum. In order to ensure freedom and independence in Europe and the third world, it is cogent to ban software patenting. (1) It must not be forgotten that in the future the world will be run by means of software, and if software patenting first has been allowed, it will be very difficult to ban it later on.

Furthermore, there is no reason to expect software patenting to ensure software development. The internet has been developed although software patenting has had ignorable influence. The operating system Linux has been developed within less than 10 years without any patenting protection, and today it is competitive to Microsoft's operating systems; software patents would have made such a development impossible, and Microsoft (or Macintosh) would not have experienced any new competitive challenge.

If the physical-chemical effect of the application of a new piece of software is in the very storage and processing of the software, it should not be patentable (as it has been the case in USA) because then all kinds of software would be patentable.

However, software is often developed to solve specific technical problems, and if such problems were solved 'technically', they would have been patentable. In order to place such technical software innovations on equal footing with 'pure' technical innovation, it seems reasonable to make an exception to the rule that software is not patentable. (2) If a new piece of software is tightly connected to solution of a specific technical problem it might be patentable. But, if the software can be used in other contexts, it cannot be patented there. (3) The reason for this is that algorithms often are so widely applicable that it will hamper software development in general and promote monopolisation if it is possible to patent any application of an algorithm.

The newness of the physical-chemical effect that is the outcome of software application in specific machinery or instruments is sometimes mostly practical, symbolic or imaginary rather than physical-chemical. Thus,

the effects of new algorithms for organising images on a monitor are in general not physically-chemically new, but mainly imaginary or symbolically new. Since such software algorithms normally have very broad applicability, they tend to hamper software development and promote monopolisation. An example is the one click patent of www.amazon.com.

One of the problems with software patenting is that software as such has not been defined clearly. The above proposal contains a definition of software and software as such. This should hopefully prevent a continual redefinition of which software patents should be patentable, and which should not. What made software patents possible in US – and to some extent in Europe too – was that software and software as such was not clearly defined. To control patenting practices, it is requisite that software and software as such is well defined.