

# Against Software Patents

Paul Klint \*

15th September 2003

As a computer scientist I am interested in (a) long term research in software engineering; (b) innovation and technology transfer via education, consultancy and start-up companies. From these two perspectives I want to contribute to the current debate on patents on software implemented inventions.

**I: Patents are intended to encourage innovation by protecting the investment made by the inventor; this does not apply to software.** Not all areas of technology can be treated equally. For the pharmaceutical industry patents seem to work, at least from an economic perspective. In the case of software, innovation goes much quicker and the innovation steps are smaller and more incremental: each innovation builds on hundreds of previous software innovations. The effect of software patents is to encourage monopolies and to discourage innovation. Strengthening software patents will weaken software innovation.

**II: A range of "bad" patents exists on trivial, well-known, techniques.** Patents exist on the treatment of overlapping windows, the generation of programs by other programs, and so on. These techniques are used daily in university research, in small and medium enterprises (SMEs) as well as in large companies. Granting such trivial patents will bring any innovation to a grinding halt. There are two reasons: (a) innovation activities will be replaced by patent search activities; (b) innovation becomes harder since "obvious" solutions will have to be replaced by more elaborate non-patented solutions.

**III: The economic effects of the patent system as a whole are unclear and differ per technology sector.** In the current practice of the European Patent Office a software invention can be patented provided that it makes a "technical" contribution. In the US the criteria are more liberal and this has opened the gate for the patenting of business processes and "trivial" innovations as mentioned above. The positive effects of the current European patent law on innovation and the economy are unclear. The effects of the proposed directive are completely unknown (although one can very well argue that only negative effects are to be expected). It is therefore unclear why the current position regarding the patenting of software implemented inventions has to be changed.

**IV: Research in many branches of science and technology depend on unpatented/open software and algorithms and will suffer from software patents.** Increased patenting of software and algorithms will slow down research in many areas. This is caused by the following effects: (a) The patenting process slows down the dissemination of new findings, since they have to be kept secret while the claim is being investigated. (b) Many patents are described in such global terms that they are hard to reproduce by others. Compare this with the rigorous descriptions in scientific publications that are intended to enable reproduction of the results. (c) Obvious, but patented, solutions are ruled out and resources are needed to find more elaborate, non-patented, solutions (that may even be inferior). Consider, for instance, research on 3-dimensional structures, from nano-structures to proteins and viruses. This area is clearly of major interest from a scientific and an economic point of view. Recently, the European Crystallographic Association has made a public statement that patents on software and algorithms will stifle the innovation in their field. Similar observations can be made about other areas of research.

**Overall conclusion** The best option is to rephrase the proposed directive in such a way that software is treated in a manner that takes into account the above considerations. What is needed, in particular, is a crystal clear definition of what constitutes a "technical contribution" in the area of software and what form of claims on software implemented inventions are acceptable. Any patents on software should be granted only for short periods (2–3 years at most). As part of the implementation of the directive, new, better and cheaper procedures will have to be set up to find "prior art" in the case of software and to simplify the opposition procedures to awarded patents.

---

\*Affiliations: (a) European Association for Programming Languages and Systems; (b) Department for Software Engineering, Centrum voor Wiskunde en Informatica (CWI, Amsterdam); (c) Programming Research Group, University of Amsterdam.