

A Compromise Proposal for the Computer-Implemented Inventions / Software Patent Controversy by Means of Substantive Patent Law

Markus Nullmeier, independent algorithm developer
markus.nullmeier@telu.de

Abstract: The present proposal argues for a novel solution of the Computer-implemented inventions / software patent controversy by defining several new classes of patentable subject matter. Based on general legal definitions of computer operations, it introduces a novel patentable entity: the composition of computer and program, applicable for uses of computers where no actual programming takes place, thereby fulfilling a perceived patentability need of the industry. Furthermore, the exclusion of computer programs by the European Patent Convention is completely re-defined in terms of 'basic unpatentable building blocks'. The regulations' details are highly adjustable and are intended to serve as a basis for further debate.

Keywords: Computer-implemented inventions, software patents, substantive patent law, European Patent Convention, sui generis rights, legal definition of computing, patent claims

Copyright statement: Copyright of the paper resides with the author. Submission of a paper grants permission to the EPIP–2007 Conference Scientific and Organizing Committees to include it in the conference material and to place it on relevant web-sites.

Acknowledgements: Stimulating discussions with Stefan Walz (German Federal Ministry of Justice), Markus Rau (Max Planck Institute for Comparative Public Law and International Law), and André Rebentisch are gratefully acknowledged.

1. Introduction

Although the rejection of the proposal for a EU directive on the patentability of computer-implemented inventions (CII directive) by the European Parliament in 2005 did calm down the respective political debate for some time, the underlying controversy as regards substantive patent law remains essentially unresolved to this day. For, the status quo of patentability in the realm of computing in Europe was deemed unsatisfactory by both the directive's proponents and the opposition to it, as both camps pushed for quite involved changes to statutory regulations of substantive patent law (European Commission, 2002; Wikipedia, 2007).

However, it seemed at the time that compromise options within substantive patent law were virtually unfeasible inasmuch a common terminology for the subject matter in question did not exist. The opponents spoke of the directive's subject matter in terms of software patents, a term common in the United States and elsewhere, while the proponents maintained the term computer-implemented inventions, to be understood as something quite different from software patents.

Thus, for the time being, patentability questions in computing will continue to be the domain of case law. In view of the present enormous relevance of computers to industrial value creation, it seems however doubtful that the evolution of case law will easily contribute to the stability of law. A few recent rulings of courts in the United Kingdom are indicative of the latter (Pearce, 2007).

2. The problem of defining patentability limits for computer programs

When the European Patent Convention (EPC) was negotiated and signed in 1973, it would have been hard to image the ubiquity of computers in 2007, having microprocessors embedded in basically every electronic gadget on the market. A 1973 vintage computer would be usually be contained a big box too heavy to lift for a single person, while a 2007 vintage computer may comfortably fit, in form of a digital hearing aid, into a human auditory canal. And yet both examples have at their heart technically much of the same kind of device: a machine capable of performing arithmetic and logical computations very quickly.

The EPC excludes, amongst other subject matter, 'programs for computers' from patentability, while narrowing the exclusions with the notorious 'as such' clause, leading to evolving interpretations that made Hilty and Geiger (2005) claim that today "the letter of the law is nothing but hollow words". Thus the lack of a common terminology during the debate about the CII directive may also hint at inherent difficulties in finding an appropriate terminology.

Coming back to the digital hearing aid, one may want to have all functionality of such an object be patentable, as formerly hearing aids did not contain computers at all and were clearly patentable. By this example we observe the almost complete convergence of (patentable) hardware and (as wished by some, non-patentable) software, prevalent in many fields of technology. Presently, elaborate hardware contraptions are often more economically replaced, retaining the very same function, by programs run-

ning on small computers embedded in specific products — why, one might argue, should this change patentability?

On the other hand, one may still want to exclude programs installed on desktop computers or bigger hardware from patentability, as voiced by the opposition to the CII directive, for economic and other reasons, thereby continuing the regulatory effect the EPC may have had in its early years (Winischhofer 2000).

3. A compromise proposal detailed in terms of substantive patent law

In view of the above, the present work proposes a twofold solution: Firstly, a new entity within substantive patent law, called *composition of computer and program*, addresses the patentability needs arising from the ever growing convergence of “classical” engineering disciplines and computer science. The term *composition of computer and program* is defined in such a way that hardware and software which is reasonably inseparable, such as the above digital hearing aid, as well as the proverbial computer-controlled washing machine and ABS breaking system, fall under this notion.

Secondly, the exclusion of programs for computers following the European Patent Convention is defined anew in detail for all cases that do not fall under the preceding paragraph. In a nutshell, only programs that are not pre-installed on a given computer and do no more than internal calculation and interactions with humans will thus be non-patentable, for example programs for everyday use on personal computers. On the other hand, any means going beyond that, such as to for example the control of industrial production equipment by software, will be clearly patentable.

The proposal sketched above has been put into statutory language; and, for reasons of simplicity, it has been detailed in terms of a seven–page amendment to the European Patent Convention. The following sections summarise its essential content.

3.1 Legal definitions for computing terminology

As a prerequisite, it is necessary to delineate an as appropriate as possible terminology comprising the basic modes of computer hardware and program operation that are to be excluded from or included in patentable subject matter.

Firstly, a computer is to be understood as a *digital* computer, namely a machine operating exclusively with information in terms of discrete values. Thereby we rule out other types of devices that are also called 'computer', usually with a distinctive adjective. Next, information processing on computers means only digital computations that are orchestrated by discrete time steps that transform one information state into the next one.

There are also input and output devices — generally of an unspecified kind — communicating digital information in and out of a computer, important special cases of which are other computers and data storage media. A set of computers connected in this way (a network) is to be considered a computer in its own right if all of them are confined into a small physical space with lengths of the order of one metre.

Programs are defined as arbitrary subsets of all of the digital information within a computer. (Note: Because of this, entities colloquially called 'data', such as word processing documents, are programs as well.) Digital information is usually and in principle, but not always and concretely, changeable by computations. Program installation equates to entering it via input devices, except in cases of mere archiving. Usage of a program is the usage of the respective computer and its associated input/output devices in dependence of said program.

Complete sources of programs are all digital information required to construct a program in the form they are edited by the program's author; external contributions to a program are those informations not edited by the program's author. This is important for the definition of *non-public programs*, these are constructed by relying on non-public, i.e., secret, technical documentation of either computer hardware or external contributions — if there is an unbroken chain of relying on non-public documentation from the hardware to the program in question.

A non-public program in terms of the above paragraph will lose this quality if all its complete sources and external contributions are available at competitive prices and are thus not considered secret.

One should note that these definitions are somehow but purposefully more general than everyday or even information technology specialists' engineering definitions of some of the same words. By this generality we hope to gain reasonable independence of the fast-changing state of the art. Here and in the following sections, it turned out to be convenient to define technical entities in a style reminding of patent claims.

3.2 Novel patentable entities: compositions of computer and program

The new term *composition of computer and program* serves to differentiate between programs for computers as a means to a specific end — such as the software within a digital video recorder — and programs that are a result of the programmability of a computer. This construction hinges on the facts that a) the playing software is installed on the digital video recorder at the time of purchase and, alternatively, that b) the construction of said software is essentially impossible without detailed technical information that is usually closely guarded as a trade secret. Conversely, programming of the video recorder by the home user (the definitions of Section 3.1 cover this colloquial term nicely) falls under neither of points a) nor b) and will thus be patent-free. A composition of computer and program is thus meant to indicate the effective absence of programmability, either because of lack of accessible documentation or because of the fact that the software was already there at point of sale.

Concerning point a), if one the following facts hold true *at the time of physical exchange* related to a purchase, a composition of computer and program is construed:

- the program is installed on the computer
- program and computer are inseparably boxed
- program and computer are transferred for a combined, single price (subsidies in either direction do not equal a single price)
- the joint purchase of program and computer enacts a discounted price.

Note that this would include the vast majority of operating systems presently sold for personal computers.

Concerning point b), any non-public program as defined in Section 3.1, with respect to the computer concerned, constitutes a composition of computer and program. However, digital information in the style of usage passwords, cryptographic locks, copyright licensing schemes, digital rights management (DRM) schemes or other artificial secrets restricting the usage of program never suffice to constitute a composition of computer and program.

It is very important to note that, in order to precisely capture the patentability only for software that acts after a fashion as a hardware replacement, all rights and prohibitions from a patent of a composition of computer and program are linked to the computer as a physical object. A patent violation may only happen after the installation of the program in question.

Thereby, a separate sale of software will not fall under possible patent restrictions. The respective liabilities should be further limited to manufacturers of hardware (these include of course subjects that pre-install software), as opposed to mere resellers. Naturally, a manufacturer disseminating non-public information leading to non-public programs, implying compositions of computer and program, will want to carefully word contracts with external software creators receiving non-public information. Otherwise, there would be no eventual compensations for patent violations depending on decisions made by the external software creator.

The complications of the last paragraph are somewhat unfortunate but indispensable for this proposal to act as a compromise. For, without this patent indemnity for software distribution regarding compositions of computer and program, it would be still feasible to sue and thereby economically threaten software vendors for the sake of legal uncertainty, even if the existence of a composition of computer and program will be easily ruled out in court.

3.3 Narrowing the limits of non-patentable computer programs

If a program is not subject to a composition of computer and program of Section 3.2, it may be called a *non-composite program*. Rather than trying to define software-controlled processes, such as blast furnaces to produce metal, that one wishes to include in patentable subject matter, we proceed by detailing patentability exclusions. They are meant to comprise all conceivable interactions of humans with computers, and of inter-computer interaction in digital networks.

All aspects of the operation of non-composite programs are exempt from patentability, inasmuch they are composed from the following building blocks:

- digital computations as by Section 3.1
- structure of digital information
- computations with information gained from patentable processes; if done outside of said processes, for which a time delay of five seconds is sufficient
- digital information representing non-patentable subject matter as per EPC, or renditions of human language
- using output devices that produce results directly perceptible by human senses
- using input devices that substitute human perception
- using input devices controlled by humans
- unsteady processes using above devices on human-perceptible time scales

- input of physical time that is accurate to the computer's time step(s)
- input of the computer's localisation up to metres
- input of true random numbers
- digital communication and protocols on networks and data storage media

Note that there is a deliberate shortage of output devices operating as actors: e.g., while the action of a force-feedback joystick is directly perceptible by human senses, the actions of the arms of an industrial robot are clearly not.

Also, any kind of simulation or digital modelling of patentable subject matter with programs meeting the above criteria is expressly excluded from patentability. Otherwise, the important functionality of computers as a support for the human brain would be seriously hampered. And, for instance, it seems relatively unreasonable to require licensing of ABS braking systems simulated within car racing video games.

Since this regulatory proposal is designed as a replacement for the 'as such' clause number 3 of article 52 EPC that allows patenting of programs for computers only in a highly contested way, desired additional patentable subject matter has to be expressly defined. Consequently, functional combinations of most of the unpatentable subject matter of this section and other patentable subject matter are to be defined as patentable, in order to enable patents for the distinctive use of computers in general technical applications.

3.4 Miscellanea

Pre-existing patents of any kind will not be invalidated by this proposal. Rather, their applicability may be re-interpreted in terms of Sections 3.2 and 3.3. In this way, any computer-related subject matter stays patentable as long as related hardware and software are sold at the same time; and it seems plausible that these are by far the most abundant cases.

An amendment to the European Patent Convention appears to be the technically most appropriate means to put this proposal into law. A version in statutory language (in German, available from the author on request) has been written mainly as an additional chapter to the EPC. However, other options, such as a renewed European directive, remain in possible principle. Political aspects of either procedure are out of the scope of the present work.

4. Related work

It has been argued (Hilty and Geiger, 2005) that a *sui generis* right for computer programs replacing possible patent protection might be created. Hollar (2005) proposed such a new protection system that combines aspects of patent and copyright. Winischhofer (2000) proposed to delete clause number 3 from article 52 EPC, without any substitution, in order to remove most computer programs from patentability.

5. Conclusions

It has been shown that detailed regulations fulfilling diverse requirements for the patentability of computer operations can be formulated, while catering for the needs of different kinds of stakeholders. The present wording aims at a consistent solution, but

the detailed regulations of Section 3 will allow for a substantial amount of adjustability. Hence it is to be hoped that the present work will help further discussions in this area or even be a step in the direction of a common language for patentability and computing.

While the present proposal has been formulated in terms of substantive patent law, it comes close to amount to a *sui generis* solution in its own right. But it also accounts for inclusion into and exclusion from patentable subject matter — an issue that would not automatically vanish in case of the establishment of an additional type of intellectual property protection coined specifically for the usage of computers.

In the future, the regulations of Section 3 should be put to the scrutiny of test cases, such as those of the UK Intellectual Property Office (2005), or existing patents that are considered relevant, in addition to input from stakeholders.

References

Commission of the European Communities, (2002). Proposal for a directive of the European Parliament and of the Council on the patentability of computer-implemented inventions. Brussels, (COM (2002) 92 final). Retrieved August 31, 2007, from http://europa.eu.int/eur-lex/lex/LexUriServ/site/en/com/2002/com2002_0092en01.pdf

Hilty, R. M., & Geiger, C, (2005). Patenting Software? A Judicial and Socio-Economic Analysis. *International Review of Intellectual Property and Competition Law*, 36(6), 615–646

Hollar, L.A., (2005). A New Look at Patent Reform. *Journal of the Patent and Trademark Office Society*, 87(9), 743-754

Pearce, D. (2007). The fallout from Aerotel/Macrossan. Retrieved August 31, 2007, from IPKat: <http://ipkitten.blogspot.com/2007/07/fallout-from-aerotelmacrossan.html>

UK Intellectual Property Office, (2005). Technical Contribution Workshop, UK Patent Office. Retrieved August 31, 2007, from <http://www.oss-watch.ac.uk/resources/techcont05.xml>

Wikipedia authors, (2007). Proposed directive on the patentability of computer-implemented inventions. Retrieved August 31, 2007, from http://en.wikipedia.org/wiki/Proposed_directive_on_the_patentability_of_computer-implemented_inventions

Winischhofer, T., (2000). *Computersoftware und Patentrecht*. Linz: PhD thesis. Retrieved August 31, 2007, from <http://www.winischhofer.net/other/patent.pdf>