Brussels, February 2005

Patenting Software vs. Free software

What should the European Union do?

Briefing Paper - by Sandra R. Paulsson[1]

Summary

A proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented invention came in 2002 (COM(2002)0092). The proposal has been adopted by the Commission and the Parliament for the first reading. In May 2004 the Council reached a political agreement by a qualified majority on a common position regarding the proposal, but has not yet adopted the directive.

Software patenting is surrounded by a swirl of uncertainty, maybe because it is a subject that is very complicated and technical by nature. This paper will highlight the principle legal dilemmas regarding the proposed directive. The goal of this briefing paper is to explain **how the current software regime in the Unite States works and compare U.S. developments to the proposed directive.** To state it bluntly, the U.S. and the E.U. have misunderstandings of the other parties' law of patents:

- Both the EU and the U.S. do not think that the other has a list of what is not patentable subject matters, but both side have it (EPC Art. 52-53 and 35 U.S.C. 102-103)
- The criteria for patentability are seen as very different on both sides of the Atlantic, but in fact they are very similar. Different labels but with the same definitions
  - In several communications within the EU it is said that the U.S. only have two or three criteria for patentability, when there are in fact four (have to fall within one of the classes of patentable subject matters, be new, useful and non-obvious, 35 U.S.C. 101-103).
  - Europe contend that the biggest differences between U.S. law and EU law is that in Europe there is a requirement of an inventive step (technical contribution) and industrial applicability in order to be patentable, and that America lack these requirements, that is not entirely true (see 35 U.S.C. 101 & 103).
  - In communications from the Commission it is said that business methods can be patented in the U.S. because it has an economical effect and that that would never be possible in Europe because here we require a technical contribution - this is not entirely true. In fact one can not get a patent in the U.S. only because it creates a tangible result, the invention has to fulfil the four criteria as well (*State Street Bank v. Signature Finance,* 149 F.3d 1389 (Fed. Cir. 1998). Further, most of these inventions can actually already be patented in Europe today.
  - In communications from the Commission it is said that in America one can patent anything, one just have to use a computer, and that this has to be avoided in the EU, but the proposal of the directive will allow that to happen here in Europe as well.
- The difference between copyright and patent is not entirely clear among SME and people. It is difficult to understand what the difference is if one does not understand what patent and copyright are and that they are complementary, not parallel.
- Mistakes regarding what the Free Software Movement and the Open Source Movement stand for, and arguments that they lack knowledge of patent, when in fact it is the contrary.
- What the real risks are for SME and consumers. Communications says that patents are too expensive for the SMEs therefore they rather use copyright, and they want to remedy that by giving them information.

The goal of this paper is not to take a stand about whether the possibility to patent software is a good or a bad thing. But, rather to give correct information about the law in America and to provoke a debate. Then we can really learn from their mistakes and be able to advance a real beneficiary balance in Europe between promoting innovation AND protecting innovation.

[1] **Sandra R Paulsson,** Dipl. Jur. (Lund University of Law, Sweden), LL.M. (SULS, Boston)**,** Trainee at the Policy Department for Economics and Science, DG 2, European Parliament, SPaulsson@europarl.eu.int & srpaulsson@hotmail.com, Phone: +32 475 229 338. The author is the sole responsible for the opinions expressed in this paper and do not necessarily reflect the position of the European Parliament.

Basic outline

Introduction

One of the main discussion topics in the information industry is the possibility of patent protection for software and business methods. In Europe, there are not only national patents, but also the relatively new European patents granted by the European Patent Office in Munich. To date, Europeans have taken a sceptical view about the patentability software and business methods. A proposed Directive would "harmonise the conditions for the patentability of inventions related to computer programs."[2] The European Commission proposed a directive in 2002 that was adopted by the Commission and then by the Parliament for the first reading in 2003. In May 2004 the Council reached a political agreement by a qualified majority on a common position regarding the proposal, but has not yet adopted it and the question is when and if the Council will adopt it.

One of the cornerstones of patent rights is to promote innovation and to give the innovator an incentive to continue, by giving him monopoly of his invention. The monopoly is limited to a term of 20 years, in order to give the innovator time to exploit his idea and benefit from it, without the risk of some individual or entity misappropriating his idea. At the end of the twenty-year term of patent protection, the innovation passes into the public domain for everyone to use. One of the purposes with this paper will be to discuss the policy-based issue of whether patenting software promotes or inhibits innovation. Another goal of this background paper is to compare and contrast European patent law with its American counterpart. Americans have recognised software patents since the 1980s and in 1998 validated the business methods patent. Europeans are considering extending patent protection to software and business methods because of developments in the U.S. patent law.

By the mid-1980s, as the consumer and business demand for software exploded, it was unclear how these intangible assets could be protected under U.S. intellectual property law. The take-off period for software patents occurred in the 1990s. Software patents grew exponentially after the United States Patent and Trademark Office (USPTO) issued Examination Guidelines for Computer-Related Inventions.[3] The USPTO coupled these guidelines with a concerted effort to recruit and train patent examiners who understood software. Prior to the 1998 case of *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*,[4] it was thought that business methods were not patentable. No single development has spurred the growth of software patents more than the *State Street* case. The Court of Appeals for the Federal Circuit stated that "the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations" was patentable subject matter.[5] The number of software and business method patents skyrocketed after the *State Street* case. The demand was so great that the USPTO instituted new policies for the examination of business method applications. The USPTO now requires a second look at applications to determine whether there are previous inventions or industry practices.

In the United States, there has been a backlash against business method or e-commerce patents. Richard Stallman, Founder of the Free Software Foundation (FSF)[6] has been critical of what he regards as dubious patents that restrain trade. Richard M. Stallman stated in a speech given in Melbourne that "the granting of patents for software ideas will be the death blow to free software

[2] World Intellectual Property Organization, What is the PCT? (visited Feb. 2, 2005) http://eurp.eu.int/comm/internal/market/en/intprop/indprop/99.htm.

[3] United States Patent * Trademark Office, Examination Guidelines for Computer-Related Inventions, 1184 U.S. Pat. & Trademark Off. Official Gazette 87 (1996).

[4] 149 F.3d 1368 (Fed. Cir. 1998).

[5] Id.

[6] See e.g. http://gnu-tr.org/ (FSF´s homepage), and http://directory.fsf.org/ (co-operation between FSF and UNESCO)

developers."[7] Stallman's model is to give away software free from proprietary restrictions (GNU/LINUX) to everyone. This approach benefit users, consumers, and is embraced in the joint effort to give every single person in the world the benefit of information communication technologies (ICT) for a low price because their ideology encourage the development of better and cheaper software.[8]

Another important aspect of this paper will be to reach an answer to whether the patentability of software will actually hurt consumers/users to get more, better, faster and cheaper access to ICT, which is the aim of the United Nations and, of course, the European Union. Another opposition is from the Open Source movement with the basic idea that source codes should be available to everyone. Also concerns are raised from small-medium sized enterprises (SME), which today rather use copyright then patent to protect software because it is cheaper and faster. SMEs' biggest concern is that they are afraid that they will not be able to compete against the big companies.

Conclusion of the main questions to be answered:
Is there any discrepancy between the EP and Commission's positions regarding patenting software? How does the legislation in the EU and in the U.S. really work? Patented software vs. free software? Will SMEs benefit from patenting software? Will patentability of software hurt consumers/users to get more, better, faster and cheaper access to ICT?

The first part of this paper will describe the main points regarding the proposed directive and the amendments made by the European Parliament in general. Next I will define the proprietary protections, notably copyright and patent through comparison between America and Europe, as well as removing some of the misconceptions. The last part will discuss the resistance of patenting software and the risks with implementing the directive and what would happen without a directive.

[7] Article in The Age, *RMS hits out at software patents,* October 8, 2004 at http://www.theage.com.au/articles/2004/10/08/1097089543132.html?oneclick=true

[8] This is one of the goals of World Summit on the Information Society (WSIS), where the possibility to use free software has been discussed to obtain access to ICT to everyone to a low price. Final Report of the Geneva Phase of the SummitWSIS-03/GENEVA/DOC/0009 (rev. 1), 2003, at p. 22, 61-62. http://www.itu.int/wsis/documents/doc_multi-en-1191|0.asp

Part I The Proposed Directive about Patentability of Computer-Implemented Inventions[9] - *why was the proposal made? Were in the process is the directive and where is it heading?*

Already in the 1997 Green paper on the Community Patent and the Patent System in Europe, there were discussions about the possibility to patent software. During 1999/2000 the discussion intensified and studies showed that European software had difficulties to compete on the market, especially against software produced in the United States.[10] The Commission invited comments and opinions on whether a harmonisation on this area was necessary and possible. The outcome of that consultation clearly showed a demand for action and harmonisation, but it was very unclear to what extend the regulations should be.[11] A proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented invention[12] came in 2002, and was aimed to harmonise the national patent laws dealing with inventions using software. These inventions can already now to some extent be patented by the European Patent Office (EPO)[13] or by national patent offices.

The proposal focuses on the necessity of including "technical contribution" into the requirements, so that not every invention that is computer-implemented could be patented. This also follows the case law of the EPO and some of the member states. The Commission wants a directive that would "by creating transparency and legal certainty, would create an environment in which [technical] innovation could be most effectively protected and fostered." The Commission stroked the possibility to patent a computer program on its own (meaning isolated form a machine), in order to avoid any misunderstandings of the scope of copyright protection and patent protection.[14] The proposal also suggest a benchmarking after three years, to keep the door open for any necessary changes and to resolve any eventual inconsistencies with the European Patent Convention.[15]

1. The First Reading of the European Parliament

The directive has gone through the first reading of the European Parliament (24/09/2003) and is still awaiting the Council to adopt or refuse the proposal, in the co-decision procedure. The Committee on Legal affairs and Internal Market took the Commissions line on patentability of software, against the position of the Greens, GUE and the proponents of "free software".[16] Once Council has taken a decision and the EP is about to make the second reading it is not possible for the parliament to make any new amendments, unless the Council rejects the proposal and ask the Commission for a new one.

[9] European Commission, The Proposed Directive on the patentability of computer-implemented invention COM(2002) 92 - C5-0082/2002 - 2002/0047(COD).

[10] STOA Information Note No 18/2000, *European Software in the world market,* available at http://www.europarl.ep.ec/stoa/publi/pdf/briefings/18_en.pdf and report from the Commission COM(2002) 92 Final, 2002/0047 (COD).

[11] PbT Consultants Ltd., *The results of the European Commission Consultation Exercise on the Patentability of Computer Implemented Inventions,* Final Report under contract number PRSW/2000/A0-7002/E/98 (2000), available at http://europa.eu.int/comm/internal_market/en/indprop/comp/softanalyse.pdf.

[12] Definition in Article 2 of the proposed Directive on the patentability of computer-implemented inventions (COM(2002) 92). "Computer-implemented invention" means "any invention the performance of which involves the use of a computer, computer network or other programmable apparatus and having one or more prima facie novel features which are realised wholly or partly by means of a computer program or computer programs."

[13] Has no communitarian authority, have only the authority given to them through the European Patent Convention by the Member states of the European Council.

[14] European Commission, The Proposed Directive COM(2002)92, Ibid.

[15] Ibid.

[16] European Parliament, The Legislative Observatory of the Directive on the patentability of computer-implemented inventions COM (2002) 92, http://wwwdb.europarl.ep.ec/oeil/oeil_ViewDNL.ProcedureView?lang=2&procid=2124

The EP voted in favour of the directive with a clear majority, after adding several amendments to the Commissions proposal.[17] The two main issues was regarding the definition of "new technical contribution" and "computer-implemented inventions", and the possibility of a "grace period" for inventors. The Committee on Legal affairs and Internal Market adopted the report by Arlene McCarthy (rapporteur: PES, UK),[18] who approved the general idea in the proposal, but limited the scope and clarified the definitions of the Directive. The main arguments regarding some of the controversial issues:[19]

1. For anything to be patentable in Europe it has to fulfil three criteria: the *invention* has to be (i) new, (ii) involve an inventive step and (iii) be susceptible of industrial application. In order to limit the scope of the directive the term "*inventive step*" also comprises the need for an "new technical contribution to the state of the art in a technical field."[20] To decide this there is a test to be used - "whether it constitutes a new teaching on cause-effect relations in the use of controllable forces of nature and has an industrial application in the strict sense of the expression, in terms of both method and result."

   Two things will be strictly non-patentable; *computer implemented business methods* and mere *algorithms*. Because the first one only contribute to the state of the art in a non-technical way and the second is inherently non-technical. However, an algorithm in a method that is solving a technical problem could be patentable, but not the algorithm itself. The discussion emphasised the need for a technical contribution *with* a technical effect in order for computer implemented inventions to be patentable, only to improve the efficiency of a data processing system would not be enough in order to be an invention. Further, the EP wants a fair use clause for operability.

   Computer-implemented inventions will follow the meaning of invention stated in the European Patent Convention and the directive will not change the status quo of the EPO's practise of granting patents.

2. Parliament asked the Commission whether or not there is a need of a Grace period, for any information about elements of the innovation that is disclosed before the application. Because, "it is necessary to ensure that the inventor will not be deprived of his/her invention, if he/she try its attractiveness to the market and the invention becomes somewhat public" especially for SME and cooperation between universities and industry.

That some of these arguments contain certain flaws will be shown in Part II.

"In the **codecision procedure**, Parliament and the Council share legislative power. The Commission sends its proposal to both institutions. They each read and discuss it twice in succession. If they cannot agree on it, it is put before a "**conciliation committee**", composed of equal numbers of Council and Parliament representatives. Commission representatives also attend the committee meetings and contribute to the discussion. Once the committee has reached an agreement, the agreed text is then sent to Parliament and the Council for a third reading, so that they can finally adopt it as law"

Title IV under the Amsterdam Treaty
Europa, Decision-making in the European Union
http://europa.eu.int/institutions/decision-making/index_en.htm

[17] 361 in favour, 157 against and 28 abstentions.

[18] Adopted the 17 June 2003.

[19] Arlene McCarthy, Report on the proposal for a directive of the European parliament and of the Council on the patentability of computer-implemented inventions (COM (2002) 92 - C5-0082/2002 - 2002/0047 (COD)), Committee on Legal Affairs and the Internal Market of the European Parliament, A5-0238/2003, 1 June 2003.

[20] European Commission, Internal market, Proposal for a directive on the patentability of computer-implemented inventions - frequently asked questions, 20 Feb. 2002, at http://www.europa.eu.int/comm/internal_market/en/indprop/comp/02-32.htm.

2. Far From Reaching an Agreement

There are some differences between, on one side the European Parliament, and on the other side the Council and the Commission. The differences mainly concern the definitions in the directive and issues relating to exceptions from patentability of computer-implemented invention.[21] The Parliament wants a wider exclusion covering the use of patented technology for interoperability and data handling. While, the Commission and the Council think that this extension goes beyond what is required to set the balance between rewarding inventors and allowing competitors to build on the inventions, and that such exclusion would harm EU competitiveness.[22] The Commission has stressed out that the European Industry need this legal environment to encourage innovation without stifling the competition, and that this Directive will contribute to foster innovation and allow those who develop innovations to profit justly from their work.[23]

The Council reached a political agreement to accept the current draft of the Commission by a qualified majority in May 2004.[24] But, the Council has not yet formally adopted the proposal even though the Dutch Presidency pushed for a decision in the end of 2004.[25] In November 2004 Poland withdrew their support of the proposal after a meeting held earlier in November.[26] During the meeting representatives from software industry and several patent lawyers agreed that the current directive makes all software potentially patentable. However, Poland also said that they would support a less ambiguous directive that does not allow the basic functions of a computer program to be patented.[27] The President of the Internet Society of Poland, Wladyslaw Majewski said that the directive accepted by the Council in May 2004 "was the biggest threat to our economical growth and the freedom of communication."[28] In January 2005 61 members of the parliament requested the Commission to send back the proposal to the European Parliament for a first reading.[29] Later that month the JURI committee in the Parliament agreed to ask the President of the Parliament to make a formal request to the Commission to send back the proposal to the Parliament for a first reading.

[21] Europa, Press Release, Patent: Commission welcomes Council agreement on Directive on computer-implemented inventions, IP/04/659, 18 May 2004, http://europa.eu.int/rapid/pressReleasesAction.do?reference=IP/04/659&format=HTML

[22] Ibid.

[23] European Commission DG Internal Market, Patents: Commission proposes rules for inventions using Software, Feb. 2002, http://www.europa.eu.int/comm/internal_market/en/indprop/comp/02-2277.htm

[24] Council of the European Union, Draft statement of the Council's reasons: Common position adopted by the Council with a view to the adoption of a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions, 11979/04 ADD 1, 24 Nov. 2004.

[25] FFII, Dutch Presidency TRIPs all over itself while modifying agenda against time, 17 Dec. 2004, at http://swpat.ffii.org/log/04/cons1217a/index.en.html

[26] Europa, IDABC, Polish Government withdraws support for EU software patents directive, 19 November 2004 and Polish Council of Ministers withdraws support for European Software Patents Directive, 24 November 2004, http://europa.eu.int/idabc/en/document/3507; The Inquirer, Poland scuppers EU software patents directive, 17 Nov 2004, http://www.theinquirer.net/?article=19736; and Lucy Sherriff, Poland halts software patent directive, the Register, 21 Dec. 2004, http://www.theregister.co.uk/2004/12/21/patents_dropped/

[27] Lucy Sherriff, Poland halts software patent directive, Ibid.

[28] Ibid.

[29] Europa, IDABC, Terms of Software Patents Debate Evolving, 14 January 2005, http://europa.eu.int/idabc/en/document/3772

Part II. Protection of Software, and other computer-implemented inventions

1. Definition of Software - what is the fuss all about?[30]

The word *invention* has neither been clearly defined in EPC[31] nor in TRIPS[32], the lack of a clear definition has lead to uncertainty of what kind of inventions that are covered by this directive. Since this has a direct impact of why this directive was initiated in the first place I will try to define it. First, some writers argue that software are already patentable under the existing rules and that the directive will not add much value to the evaluation, while others argue that it is impossible for software to be patented today and want a clearer legislation. Second, some writers argue that this directive only cover software, while others argue that the directive comprise so much more and that software is only a part of it.[33] In order to try to pinpoint what is covered in the term computer-implemented[34] inventions I have tried to find the answer through already defined terms. The definition of computer-implemented inventions in the proposed Directive can be found in Article 2, the term means any invention implemented on a computer or similar apparatus, which is realised by a computer program.[35]

Software[36] = a computer program which are stored and executed by computer *hardware*[37]. There are two types of software:

- *system software*, which includes the operational system (e.g. Windows 95, UNIX, MacOS) that enables the computer to function.
- *application software*, which includes programmes that perform specific tasks, e.g. word processor, databases and spreadsheets.

The definition of software is closely approximates the subject matter of computer-implemented inventions. With the definition of "implemented" as a method or a theory put into practice, then computer-implemented invention would just simply be software. Since the "implemented invention" according to the definition in the directive is "any invention implemented on a computer" and the word "implemented" is a kind of method, this would actually be an "application software" - a program with a specific task (meaning a sort of method/theory put into action) that can not be operable by itself, but only through the realisation of a "system software" (meaning the operational system). This would mean that the term in itself does not comprise anything more that software.

[30] My conclusion on this point might not agree with others point of view of this and I welcome and encourage any good counterarguments, but as long as there are no other indications of the contrary I stand behind my theory 100 percent.

[31] The closest to a definition is in Art. 52 (1) EPC, "European patents shall be granted for any inventions which are susceptible of industrial applications, which are new and which are not obvious."

[32] The global minimum standard for protection of intellectual property rights is in Section II.5, Art. 27(1) TRIPS that only states that "patents shall be available for any inventions, whether products or processes, in all fields of technology, provided that they are new, involve an inventive step and are capable of industrial application..."

[33] Is often referred to as "software and *other* computer-implemented inventions."

[34] I*mplemented* can be described in the computer world to be "putting a system, process or practise into operation, from concept to design, to be realised through a computer." WordNet 2.0 Search, defining "implement", at http://www.cogsci.princeton.edu/cgi-bin/webwn?stage=1&word=implemented.

[35] Article 2 in the proposed Directive COM (2002) 92 "any invention the performance of which involves the use of a computer, computer network or other programmable apparatus and having one or more *prima facie novel* features which are realised wholly or partly by means of a computer program or computer programs." The European Parliament wants to remove the italic words, Arlene McCarthy, Report, Ibid., Amendment 14 of Article 2, point (a).

[36] Defining "software" through google.com e.g. at http://www.getnetwise.org/glossary.php; http://www.microsoft.com/office/newtosite/glossary.asp; and http://iet.ucdavis.edu/glossary/.

[37] Hardware = physical components of a computer including the keyboard, monitor, disk driver, and internal chips and wiring, e.g. at http://www.bpl.org/kids/Glossary.htm.

However, the risk of exclusively using the word software could lead to exclusion of new closely related technical inventions, therefore the word computer-implemented invention could maybe serve a broader purpose in the future.

The definition in the proposed directive do not really give anymore guidelines then what is already stated in EPC and TRIPS, therefore it will be left to either ECJ to interpret the definition or for the national courts to decide how to use the definition in the light of their national legislation, if the directive is adopted.

2. Proprietary protection of software

Commentators often confound the distinction between the law of copyrights and patent law. Copyright law protects expression, whereas patent law provides protection against the infringement of patent claims. In the field of software law, copyright and patent law aim at different means of protecting these intangible assets. In the United States, software copyrights were recognised before software patents.

Below there will be a synoptic discussion of main features of patent and copyright law in Europe and America. In Europe patent and copyright are mainly protected under national law, and courts individually interpret international agreements such as the Convention on the Grant of European Patents (EPC) and the TRIPS.[38] In the United States, these proprietary interests were first recognised under Article I, §8 cl. 8 of the U.S. Constitution. The Copyright & Patent Clause states: "Congress shall have the power to promote the progress of science and useful arts." In the United States, the law of patents and copyrights are governed by federal statutes: The U.S. Copyright Act of 1976 that is located in Title 17 and the U.S. Patent Act that is found at Title 35. Both statutes have been amended in order to update the law and comply with international intellectual property treaties. The U.S. Copyright Act was amended, for example, to comply with the 1998 WIPO Treaties.

2.1. Copyright

The term "copyright" is in itself a description of what is protected - the author's exclusive right to control the reproduction and protect against copying. Copyright do not extended to ideas, procedures, methods of operation or mathematical concepts as such.[39] The principal aim is to derive

[38] The General Agreement on Tariffs and Trade 1994 (GATT) is an international agreement that sets minimum standards for the protection of Intellectual Properties and is an agreement by the World Trade Organisation (WTO). Most of the WTO agreements are the result of the 1986–94 Uruguay Round negotiations, signed at the Marrakech ministerial meeting in April 1994. The Final Act signed in Marrakech is like a cover note, with everything else attached to it. One of them is the agreement establishing the WTO (the WTO Agreement) with several annex. For example the agreements on goods, services, and intellectual property. Annex 1c on intellectual property is more know as the **Trade-Related Aspects of Intellectual Property Rights (TRIPS**) agreement. WTO legal texts, http://www.wto.org/english/docs_e/legal_e/legal_e.htm#TRIPs. The USA signed GATT 1 January 1948 and most of the European countries had also signed GATT. WTO, The 128 countries that had signed GATT by 1994, http://www.wto.org/english/thewto_e/gattmem_e.htm, The USA and the European Communities became members of the WTO on 1 January 1995. WTO, Members and observers, http://www.wto.org/english/thewto_e/whatis_e/tif_e/org6_e.htm

The Conventions by World Intellectual Property Organisation (WIPO) are also respected, notably the **Paris Convention for the protection of Industrial property of 1883** and the **Berne Convention for the Protection of Literary and Artistic Works of 1886**, both revised several times. They are also partly incorporated, through Article 2.1 and 9.1 in the TRIPS Agreement. The WTO, Other Intellectual Property Conventions, http://www.wto.org/english/tratop_e/trips_e/intel4_e.htm

[39] TRIPS Article 9 (2) and has historical implications from John Locke's natural law theory.

benefits of the labours of authors to the public and to enhance the public welfare, rather then to economically benefit the author. The duration of protection is today the lifetime of the author + 70 years.[40]

Under America Federal law Copyright Act of 1976 an "original work of authorship" is protected as soon as it is "fixed in a tangible meaning of expression."[41] Originality and fixation are the two most important pre-requisites for copyright protection.[42] **Originality** means that the work must have been independently created with a minimum degree of creativity, however an idea is not protected. This means that if the idea is so intermingled with the mean of expression then it might not be copyrightable (idea/expression Dichotomy) this is up to the court to decide.[43] **Fixation**, the embodiment of the idea/expression is necessary, since copyright do not protect ideas. For example, a description of a machine could be copyrighted, but this would only prevent others from copying the description; it would not prevent others from writing a description of their own, or from making and using the machine.[44]

Infringement occurs when someone violates one of the exclusive rights of the authors: reproduction, adaptation, publication, performance and display.[45]

2.1.1. Copyright Protection For Software

A computer program can be protected in America as a "set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[46] If the program is considered to be a procedure, process or a system it is not copyrightable, but it might be protected under patent law. In *Computer Associates International, Inc., v. Atlai, Inc.*,[47] the question how far copyright should go to protect non-literal elements of a program was addressed. The two programs at issue did not have the same program codes, but were very similar. The court developed a three-stage analysis: abstraction, filtration and comparison. In the first stage the structure of the programs was dissected on each level of abstraction in it. Then, the court filtered out each element of the program that was just for external factors (mechanical specifications) and the efficiency of the program, as well as elements taken from the public domain. In the last stage the remaining elements should be compared. The court found that defendant's program contained protectable elements similar to plaintiff's program, but that the similarities were not sufficiently material to the overall program that any infringement could be found.

Computer programs are already seen as literary work in the meaning of the Berne Convention in the EU and is protected under copyright law as long as the program is original, which means that it is

[40] The work will enjoy this protection if the author is known. Work made for hire, anonymous or pseudonymous work have 120 years of protection from the creation or 95 years from publication, whichever is shorter in the US, 17 U.S.C. 302. While this protection is the same for all copyrights in Europe - 70 years from publication, Article 1 and 7 Council Directive 93/98/EEC of 29 October 1993 harmonising the term of protection of copyright and certain related rights.

[41] Meaning on paper, on a dvd, on a painting, etc.

[42] 17 U.S.C. 102 (a).

[43] The USPTO, Copyright, at http://www.uspto.gov/main/profiles/copyright.htm

[44] Ibid.

[45] 17 U.S.C. 106 and Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society. In Europe the moral rights of the work are also protected, as an extension of the artist's personality that should not be intermingled with. In the U.S. this right is not clearly recognised, but there is a prevision about it in order to be in compliance with the TRIPS Agreement which refers back to the Berne Convention and "provides for [] the right to claim authorship of the work and the right to object to any mutilation or deformation or other modification of, or other derogatory action in relation to, the work which would be prejudicial to the author's honour or reputation." Article 6bis Berne Convention at http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html

[46] 17 U.S.C. 101.

[47] 982 F.2D 263 (2d Cir. 1992).

the author's own intellectual property.[48] The governing Directive clearly states that the underlying ideas and principles are not protected under copyright law.[49]

2.2. Patent

Patent protects inventions and it gives the inventor “the **right to exclude** others from making, using, offering for sale, selling or import” their invention. The trade-off to be given this monopoly is the shorter timelimit, 20 years instead of life+70 years as for copyright. One misconception about American patent law is that everything in the world can be patented.[50]

2.2.1. The protection in Europe

Europe has a centralised system for patent applications at the European Patent Office (EPO). The EPO was the outcome of the European countries' collective political determination to establish a uniform patent system in Europe[51] and is the executive branch of the European Patent Organisation. The European Patent Organisation was established by the Convention on the Grant of European Patents (EPC) signed in Munich 1973.[52] EPO receive more and more patent application, for example during 2004 EPO received 177,500 applications, which was an increase with 14%.[53] Generally, the national legislation of all member states should be consistent with the EPC, but since the national courts are the top interpreters of the law the outcome can be very different in practise and there is no unifying binding structure for those courts when it comes to enforcement and infringement questions.[54] Inventors can also apply for patents at their national patent office. Regardless of which option the inventor takes, national legislation will apply in the end. Which means that the legal situation today is quite unclear in Europe.

In Europe patentable invention have to be "susceptible of industrial application, which are new and which involve an inventive step."[55]

- *Industrial application* can be considered if the invention "can be made or used in any kind of industry."[56]
- *Novelty*, means that the invention "does not form part of the state of the art,"[57] which means that the invention can not already exist.
- *Inventive step* can be considered "if, having regard to the state of the art, it is not obvious to a person skilled in the art."[58] In order for computer-implemented inventions to involve an

[48] Article 1 Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs. Is also an aim laid down in Article 151 EC-Treaty "Action by the Community shall be aimed at encouraging cooperation between Member States and, if necessary, supporting and supplementing their action in the following areas []artistic and literary creation, including in the audiovisual sector."

[49] Article 1 Directive 91/250/EEC.

[50] R. Hart, P. Holmes & J. Reid, Study Contract ETD/99/B5-3000/E/106 "The Economical Impact of patentability of computer programs report to the European Commission DG Internal Market, (2000), also at http://www.europa.eu.int/comm/internal_market/en/indprop/comp/studyintro.htm.

[51] EPO, Information about the EPO, http://www.european-patent-office.org/epo_general.htm

[52] The Organisation's Administrative Council, composed of delegates from the contracting states supervises the activities of the EPO. EPO, Brochure: Information about the EPO, http://www.european-patent-office.org/epo/pubs/brochure/general/e/epo_general.htm

[53] EPO Press release, Further expansion of the European Patent System - "Patents are the backbone of the knowledge economy in Europe", 3 Dec. 2004, http://www.european-patent-office.org/news/pressrel/2004_12_03_b_e.htm.

[54] European Commission, Internal market, Frequently asked questions, Ibid.

[55] Article 52 (1) EPC, and is also stated in Article 27 (1) of TRI¨PS with a footnote declaring that the terms "inventive step" and "capable of industrial application" may be synonymous with the terms "non-obvious" and "useful" respectively,

[56] Article 57 EPC.

[57] Article 54 EPC.

[58] Article 56 EPC.

inventive step it must "in addition make a new technical contribution to the state of the art, in order to distinguish them from pure software."[59]

Further there are some subject matters that are not seen as inventions in the sense of patentability, some of these are discoveries, mathematical methods, schemes, rules and methods for performing mental acts, playing games or doing business, and program for computers as such.[60] Because the traditional view was that patent was only applied to concrete applications of natural science (technical inventions) and since software was seen as covering abstract ideas they were not patentable.[61] However, more than 30,000 software-related patents have already been granted by the EPO since 1978, because they have been seen as excluded from the non-patentable subject matter under "computer program as such" in Article 52 (2) EPC.[62] Many of these patents have been granted for devices and processes in technical areas, and today more digital data processing, data recognition, representation and information handling are granted patent.[63] However, these inventions must also fulfil the criteria of being new and non-obvious. This outcome has led to practical differences on national level and the risks for nourishing even more diversity among the European countries can come true in the absence of a harmonisation.

2.2.2. The Protection in America

For over 200 years, the United States Patent and Trademark Office (USPTO) has granted patents to promote the progress of science[64] and is today a federal agency under the Department of Commerce.[65] During 2004, the USPTO granted 187,170 patents, including **169,296 utility (inventions)**,[66] 16,533 design, and 998 plant patents.[67] Since 1790, over seven million U.S. patents have been granted and the USPTO receives about 300,000 applications each year.[68] The USPTO does not have jurisdiction over infringement questions, the enforcement, nor over matters relating to the promotion or utilisation of patents or innovations, the Federal Courts decide these matters.[69]

As a main rule "anything under the sun that is made by the hand of man"[70] can be patented, as long as the invention is new, useful[71] and non-obvious,[72] as well as falls within one of the classes of patentable subject matters.[73]

1. The words "process, machine, manufacture, or composition of matter" in the section embodies the criteria of proper subject matters.[74] Interpretations of the statute by the courts have defined the limits of the field of subject matter that can be patented, thus it has been held that the laws of

[59] European Parliament, The Legislative Observatory, Ibid.

[60] Article 52 (2) EPC, there are also exceptions in Article 53 (contrary to ordre public, plant or animal varieties). In accordance with Article 27 (3) TRIPS, allowing members to exclude certain subject matters from patentability.

[61] FFII, Software Patents in Europe: A Short Overview, http://swpat.ffii.org/log/intro/index.en.html.

[62] European Commission, Internal market, Frequently asked questions, Ibid.

[63] Ibid.

[64] As stated in Article 1, Section 8 of the United States Constitution.

[65] US PTO, Our Business: An Introduction to the USPTO, http://www.uspto.gov/web/menu/intro.html

[66] This is the group where software and business methods are included.

[67] USPTO Releases FY 2004 Performance and Accountability Report, http://www.uspto.gov/web/offices/com/annual/2004/index.html.

[68] Ibid.

[69] USPTO General Information Concerning Patents, http://www.uspto.gov/web/offices/pac/doc/general/index.html.

[70] *Diamond v. Chakrabarty*, 447 U.S. 303 (1980).

[71] 35 U.S.C. 101-102, according to Article 27 (1) TRIPS.

[72] 35 U.S.C. 103, according to Article 27 (1) TRIPS.

[73] 35 U.S.C. 101.

[74] "Whoever invents or discovers any new and useful *process, machine, manufacture, or composition of matter*, or a new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of **this title.**" 35 U.S.C. 101. The proper subject matters are also listed on U.S. Patent and Trademark Office homepage www.uspto.gov. The bold text ("of this title") means that one can not read this section all alone, but has to be read in the light of all the other sections in the whole title/law.

nature, physical phenomena, and abstract ideas are not patentable subject matter.[75] "Process" is defined as a process, act or method, and primarily includes industrial or technical processes.[76]

2. *Novelty*, means that the invention can not have been known or used by another in the US, or been patented, published or used in America or elsewhere more than one year prior to the date of the application of patent in the US.[77] Which means that if the inventor makes the invention public he has to file for patent within a year from the publication in order to secure the right to patent,[78] this is what is refereed to as the **Grace Period**.
3. *Useful* means that the subject matter has to have a useful purpose and also include operativeness (a concrete tangible result),[79] which means that a machine that does not operate to perform the intended purpose is not useful.[80]
4. *Non-obvious,* means that even if the patent sought is not exactly shown by prior art and have some slight differences it can still be refused because the difference could be too obvious. This means that the invention has to be sufficiently different from what has been done before so "it may be said to be non obvious to a person having ordinary skill in the area of technology related to the invention."[81]

2.2.2.1. Software Patent

Patenting software had been possible in the U.S. since the Supreme Court forced the USPTO to change its position in the 1980s. Before the USPTO had not granted any patents on computer programs or inventions relating to computer programs, because it saw those as mere mathematical algorithms, and not as processes or machines.[82] In *Diamond v. Diehr*,[83] the Supreme Court ordered the USPTO to grant a patent on an invention even though computer software was utilised. The invention used a computer software to calculate and control the heating time for rubber in order to be best "cured".

The invention did not only explain the computer program but also the various steps regarding the heating of the rubber. Therefore, the Court held that the invention was not merely an algorithm but was a **process** for moulding rubber and one had to look at the invention **as a whole**,[84] even if the only new feature was the timing process controlled by the computer. After this case in 1981, the USPTO was left trying to figure out when and if it was a patentable invention and when it was merely an unpatentable algorithm. This has since then increasingly been dependent on the formulation of the patent applications by the patent lawyers. The USPTO felt overloaded with

[75] *Diamond v. Diehr*, 450 U.S. 175 (1981). This is further defined in the US PTO Examination Manual Title 706.03(a) Rejections under 35 U.S.C. 101, and Title 2105 Patentable Subject Matter, http://www.uspto.gov/web/offices/pac/mpep/mpep.htm.

[76] 35 U.S.C. 103 (b)(2); USPTO General Information Concerning Patents, Ibid.

[77] 35 U.S.C. 102 (a)-(b). See *Pennock v. Dialogue*, 2 Pet. 1, 24 (1829) (holding that an inventor who makes a discovery public or allows other to use it freely loses the right to patent it unless it had been applied for before such use)

[78] USPTO General Information Concerning Patents, Ibid.

[79] Michael Guntersdorfer, Software patent Law: United States and Europe Compared, Duke L. & Tech. Rev. 006 #9 (2003).

[80] US PTO, http://www.uspto.gov/web/offices/pac/doc/general/index.html#novelty

[81] Ibid., "A patent may not be obtained though the invention is not identically disclosed or described as set fourth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." 35 U.S.C. 103 (a). See *Hotchkiss v. Greenwood*, 52 U.S. 248 (1851)(holding that the porcelain door knob was obvious in the light of prior art metal knobs); *Cuno Eng'g Corp. v. Eutomatic Devices Corp*., 314 U.S. 84 (1941) (holding that a new device must reveal a "flash of creative genius"); and *Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp.*, 340 U.S. 147 (1950) (holding that the patentee had added nothing to the knowledge, but had merely brought together parts of prior art and claimed patent for the compilation of them).

[82] Bitlaw, The History of Software Patents, http://www.bitlaw.com/software-patent/history.html

[83] 450 U.S. 175 (1981).

[84] Ibid., at 187 and 192-193.

applications on patenting software, and in 1995 it created some guidelines for the examination of computer-implemented inventions.[85] There is no special patent class for software, but what is patentable is a process or a machine, therefore the guidelines try to assist with that distinction.

2.2.2.2. Business method

Business methods patents and software patents are within the group of utility patents, along with inventions, chemical formulas, processes and other discoveries. It is seen as a process, since it is not a physical object like a mechanical invention or chemical composition.[86] Patentability of Business methods were first described in *State Street Bank v. Signature Finance,*[87] where the processing system of taking data and predict the final share price through mathematical calculations was patentable since it was useful, concrete and with a tangible result. This was the narrow outcome of the case, while the broad outcome opened up the door for patenting specific, practical application of algorithm-based software. The *State Street* court held that the business method exception had never existed as such, but that prior business methods inventions always had been denied patent protection, because they failed to fulfil the other criteria for patentability.[88]

One common misconception is the European belief that one can patent a method in America for having a certain effect, like an economical one, by the mere fact that the invention uses a computer and that no specific technological contribution is needed.[89] This is wrong, because the reason why it is patentable has nothing to do with the economical effect, one has to look at the **whole method in itself** to see if it contains a technical contribution and thus patentable. The *State Street* court also held that "patent laws is intended to protect any method, whether or not it requires the aid of a computer, so long as it produce a 'useful, concrete and tangible result'", as well as fulfilling the criteria for patentability, notably fall within the classes of patentable subject matters, be new, useful and non-obvious.[90]

Before the *State Street* case American courts had refused to patent business methods in a strict sense because it was seen as an abstract idea, an unprotectable algorithm.[91] But since this case several Internet-related patents has been granted by the USPTO, some famous examples are the "1-Click" system for on-line buying, where Amazon's method allows repeat customers to bypass address and credit card information because Amazon can directly access that information;[92] and online auction system called "name your price" or "reverse auction," where consumers name the price they are willing to pay and the first willing seller gets the deal.[93] During the last years America has tried to narrow down the possibility to patent business methods in accordance with the

[85] Bitlaw, The history of Software Patents, Ibid. United States Patent * Trademark Office, Examination Guidelines for Computer-Related Inventions, 1184 U.S. Pat. & Trademark Off. Official Gazette 87 (1996).

[86] NOLO, Internet Business Method Patent, 2004 at http://www.nolo.com/lawcenter/ency/article.cfm/ObjectID/C2DBFF26-7097-4B7B-AE36DA00499851EE/catID/2D212B35-B211-4FD4-B46D84A00C15CEBF

[87] 149 F.3d 1389 (Fed. Cir. 1998).

[88] *State Street case*, Ibid. at 1375.

[89] "The real difference between the USA and Europe is that in Europe the invention has to be of a technical character whilst in the USA *the mere fact that the invention uses a computer/software makes it of the technological arts*, if also useful, concrete and tangible results are provided.[](Of course in Europe, the USA and Japan whatever is being patented has also to be new and inventive)." R. Hart, P. Holmes & J. Reid, Study "The Economical Impact", Ibid. "In the U.S. on the other hand, *a patentable invention must simply be within the technological arts. No specific technological contribution is needed*. The mere fact that an invention uses a computer or software makes it become part of the technological arts if it also provides a "useful, concrete and tangible result." European Commission, Internal market, Frequently asked questions, Ibid.

[90] *State Street case*, Ibid. at 1375.

[91] NOLO, Internet Business Method Patent, Ibid.

[92] This business patent was granted in September 1999 (U.S. Pat No. 5,960,411).

[93] U.S. Pat. No. 5,794,207.

State Street case[94] and the USPTO has also developed an extensive Guideline for computer-implemented business method inventions in order to make the examination clearer.[95]

2.2.3. Comparing the American and the European patent system

There are three main differences between European and American patent law. First, the United States awards patents to the first person to invent, whereas the rest of the world awards patents to the “first to file” the patent application.[96] A second difference is that in the United States, the statute permits a "grace period," meaning that the inventor can publicise or publish information about an invention for a one-year period prior to filing. In Europe, the invention would be lost if such publicity were given. A third difference is that Europeans will not patent inventions that are immoral, illegal, etc.[97] The U.S. does not take that dimension into account, as long as the statutory requirements under the Patent Act are fulfilled, a patent is granted.

One major misconception of the American law compared with the European is the requirement for patents. In Europe there are several authors explaining that in the U.S. there are only two criteria for patentability, notably novelty and non-obviousness, and that the U.S. do not have a list of excluded subject matters.[98] It is interesting to see that some of these misconceptions are the same on both side of the Atlantic.[99] One of the most critical issues during the draft of the directive was regarding the "technical requirement" that Europe tries to emphasis and argues that the U.S. law is missing. However, this is not entirely true.

➔ First of all, there are four criteria for patentability in the US: be within the classes of patentable subject matters, be new, useful and non-obvious, as explained above (35 U.S.C. 101-103).

➔ Second, in Europe the invention must be **industrial applicable**, Article 57 EPC, which means that the invention must "be made or used in any kind of industry." This is similar to the criterion **useful** in America, even if this comprise more about operativeness than having a technical feature as the criterion industrial applicable include.

➔ Third, according to the Commission the central requirement for the proposed directive is that the computer-implemented invention, in order to be patentable, "has to make an contribution in a technical field that is not obvious to a person of normal skill in the field."[100] If the invention do not make a technical contribution it would be considered to lack an **inventive step** and not be patentable in Europe.[101] The requirement of technical contribution is supposed to be biggest difference between American and European patent law when it comes to computer-related inventions.[102]

[94] See *WMS gaming, Inc. v. Int'l game tech.*, 184 F.3d 1339 (Fed. Cir. 1999).

[95] USPTO, Formulating and communicating rejections under 35 U.S.C. 103 for applications directed to computer-implemented business method inventions, http://www.uspto.gov/web/menu/busmethp/busmeth103rej.htm.

[96] International Chamber of Commerce, Current and Emerging Issues Relating to Specific Intellectual Property rights (visited Mar. 3, 2003), http://www.iccwbodot.org/hom/intellectual_property/current-emerging/roadmap.asp.

[97] Article 53 (a) EPC.

[98] See e.g. European Parliament, The Working paper for the Patentability of computer programs - discussion of European-level legislation in the field of patents for software, Legal Affairs Series, April 2002, p. 9-11, http://www.ivir.nl/publications/other/softwarepatent.pdf.

[99] Michael Guntersdorfer, Software Patent Law, Ibid.

[100] Article 56 EPC. European Commission, Internal market, Frequently asked questions, Ibid.

[101] European Commission, Internal market, Frequently asked questions, Ibid.; European Parliament, The Legislative Observatory, Ibid. Recital 12 in the Proposed directive COM(2002)92, Ibid.

[102] European Commission, The Proposed Directive COM(2002)92, Ibid. p.5; and European Commission DG Internal Market, Patents: Commission proposes rules for inventions using Software, Ibid.

According to a report given to the Commission and caselaw from the EPO Art 27 and 29 EPC are supposed to identify that "technical character" is the essential requirement: [103]

- Art 27 (1) "The description shall (a) specify the technical field to which the invention relates..."
- Art 29 (1) "The claim shall define the matter for which protection is sought in terms of the technical features of the invention"

However, this is pretty much what is said about the filing requirements/rules of the U.S.PTO. Saying that one has to classify and describe the technical character of the invention.[104] Further, the report explain, in the practice note to the guidelines of the EPO technical character/contribution is defined to be "if [the program for the computer] cause, when run on a computer, a technical effect which may be known in the art but must go beyond the "normal" physical interactions between program and computer." This is not much different from America's position to patent computer programs as shown above.

An invention has to be "technical" in order to be patentable in America as well. It is clear that these kinds of inventions are more or less technical by nature, illustrated by a common definition of invention: a "technical solution to a problem." Further the definition of the American criteria "**non-obvious**" means that it should "be nonobvious to a person having ordinary skill in the area of *technology* related to the invention."[105] Which is also compatible with TRIPS where the text explicitly says that the criteria "inventive step" also can be called "non-obvious".

Therefore it seems as if the criterion "nonobvious" in America[106] is more or less the same as "inventive step" in Europe.[107]

➔ Fourth, it is said that the directive will not patent "pure" **business methods**, meaning a method without a technical character,[108] this would not be possible in America either. Additionally, the proposal is not supposed to grant patents for business methods as they do in America.[109] However, "national courts have decided that there is no reason why a patent should not be granted for a machine programmed to carry out some technical functions, or technical process carried out using a computer or similar machine [...] the proposed Directive has followed this approach."[110] Which is exactly what the American court in *Diamond v. Diehr* stated when they first allowed software to be patented.[111] The *Diehr* court held that the whole *technical process* including a computer program was patentable, meaning that the invention was patentable because it was seen as a "**process**." This means that the practise of EPO to grant computer programs as processes would not prevent it from granting patents for business methods, the caselaw of the Technical Board of Appeal of the EPO has rather shown the opposite.

Even if an invention could be seen as excluded subject matter under Article 52 (2) EPC, inventions has nevertheless been patentable if it is directed at a technical process, as long as no protection is sought for the excluded matters as such.[112] Viacom applied for a patent at the EPO in 1979 for a

[103] R. Hart, P. Holmes & J. Reid, Study "The Economical Impact", Ibid. p.13.

[104] 35 U.S.C. 112, "The specification shall contain a written description of the invention, and of the manner and process of making and using it, [and] particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention." Is also further explained in the US PTO Examination Manual Title 2106 Patentable Subject Matter - Computer-Related inventions, http://www.uspto.gov/web/offices/pac/mpep/mpep.htm.

[105] See the definition of "non-obvious".

[106] 35 U.S.C. 103

[107] Art 52 EPC

[108] See e.g. European Commission, Internal market, Frequently asked questions, Ibid.

[109] See e.g. European Commission, Internal market, Frequently asked questions, Ibid.; European Parliament, The Legislative Observatory, Ibid; and European Commission, The Proposed Directive COM(2002)92, Ibid.

[110] European Commission, Internal market, Frequently asked questions, Ibid.

[111] See *supra*.

[112] *In re Vicom Sys., Inc.*, 1987 O.JEP.O. 14, 19 (Tech. Bd. APP. 1986)

"method and apparatus for improved digital image processing," which it had already obtained a patent for in America the previous year. The Examination division at the EPO refused the patent because the method was either just a mathematical method or lacked a technical feature, while the apparatus was not seen as prior art.[113] However, the Technical Board of Appeal granted patent and held that "even if the idea underlying an invention may be considered to reside in a mathematical method, a claim directed to a technical process in which the method is used does not seek protection for the mathematical method *as such*."[114] This is the same kind of argumentation that the *Diehr* court had. There are several other cases by the Technical Board of Appeal that mirrors the outcome of the American caselaw, for example granting patent for business methods with technical features,[115] require the examination to look at the invention as a whole,[116] and that computer programs are patentable because they limit a general purpose computer to a specific purpose.[117]

It seems as if the biggest difference is the way Europe and America have formulated their legislation. In America the law is stated in a "positive way," telling what is allowed, while Europe makes it in a "negative sense" saying what is prohibited or not allowed - and using "unless" to state what is allowed. Further, most business methods granted in America would also be patentable at EPO, because it is seen as a process.[118] Thereby, I contend that there is actually not a very big difference between the way American law look at patent for software (computer-implemented inventions) including business methods, and the way Europe looks at it.

➔ Fifth, there are **non-patentable subject matters** in both the U.S. and Europe as shown above, with the difference that in Europe they are stated in Article 52 (2) and 53 EPC while in America they are both stated in the statute (35 U.S.C. 101) and have been developed through caselaw (e.g. *Diamond v. Diehr*).[119]

2.2.3. Comparing copyright and patent protection

None of these protections actually fits the purpose of protecting software in the fast development of technology. Moore's law has embodied the fast development of the IT-industry.[120] However the

[113] Further that normal implementation of method would constitute a computer program, which is a non-patentable subject matter. *Vicom*, Ibid. at 14-15.

[114] *Vicom*, Ibid. at 19.

[115] *In re Sohei*, 1995 O.J.E.P.O 525 (Tech. Bd. APP. 1994); with similar outcome to the *State Street* case in America. The Technical Board later limited the scope of *In re Sohei* with *In re Pension Benefit Sys. P'ship*, 2001 O.J.E.P.O 441 (Tech. Bd. App. 2000) (holding that business methods as such are not patentable and that the mere addition of a technical feature to an otherwise non-technical method did not change that, however a technical invention do not lose its patentable status if non-technical features are added)

[116] *Siemens A.G. et al. v. Koch & Sterzel Gmbh & Co.*, 1988 O.J.E.P.O.19 (Tech. Bd. App. 1987) (holding that it is unnecessary to weigh up the technical and non-technical features, because if the invention uses technical means then the possibility to patent is not excluded and that one should look at the invention as a whole); similar outcome to *Diamond v. Diehr* in America.

[117] *In re Dai Nippon Insatsu kabushiki Kaisha*, no. T 0605/93 (Tech. Bd. App. Jan. 20, 1995); similar to *In re Apparat*, 33 F.3d 1526 (Fed. Cir. 1994) in America.

[118] See also e.g. R. Hart, P. Holmes & J. Reid, Study "The Economical Impact", Ibid. Stating that "[i]n all three jurisdictions [Japan, America and Europe] business methods are patentable subject to the requirements indicated above. That the U.S. does not require the invention to be of a technical character means that the restrictions on patenting of business methods are negligible; for almost all business methods, limiting patents on them to when use of a computer/software is involved, does not reduce the value of the patents. (Indeed it may be possible to patent business methods in the States without such a limitation.) *But it is also important to realise that many computer-implemented business method inventions are of a sufficiently technical character to be patentable in Europe and in Japan*."

[119] This is a natural outcome since America is a common law based country ("Judge-made law") and Europe has a civil law approach (with civil codes).

[120] Meaning "the number of transistors per square inch in integrated circuits had doubled every year since the integrated circuit was invented." Gordan E Moore (co-founder of Intel) predicted that this trend would continue for the foreseeable future. During the last years the pace slowed down a bit, but data density has doubled approximately every 18 months,

software industry has not followed the same fast development as data density and hardware, and has developed a little bit slower.[121]

Copyright - is too long (lifetime + 70 years), but infringement only occurs if the right to copy has been violated. When it comes to software it is the source code[122] that can be protected by copyright because it is there the original expression by the author is. Therefore, it is very hard to infringe upon the programmers source code unless there is direct copying of the content. Which means that copyright do not give that much of a protection for programmers because one feature of a computer program can in most cases be written in several different ways. This means that two similar working/looking programs can have completely different source code, but still not infringe upon one another.

Patent - is good for innovation because it gives the owner of the patent an exclusive opportunity to generate income, which gives the incentive to invest in research and development.[123] Businesses often use the income from patented innovations to fund future once. Patents can also be used as security to obtain loans and to negotiate licensing agreements to use others' patents.[124] Patent also gives other inventors the right to obtain the information of the inventions, because of the disclosure requirement of the application process.[125]

However, patent monopolise **ideas**: most software are based on old software and the improved once has been improved thanks to the ability to debug them. Statistically, there are normally between one and five undetected defects, coding errors, per one thousand lines of codes in software released to the public.[126] One software that most users of normal PC are familiar with is "Microsoft Windows 2000," this program is an admittedly example of very bad code quality and contains about forty million lines of codes with fifteen errors per thousand lines - which means about six hundred thousand bugs.[127] Today Microsoft has debugged thousands of those,[128] but that was not their only program containing bugs, this is common for all Windows programs and computer programs in general.[129] Compare this to an air traffic control system involving millions of lines of codes and an airline that contains several million up to a billion lines of codes in them to function.

and this is the current definition of Moore's Law, which Moore himself has blessed. Webopedia, http://www.webopedia.com/TERM/M/Moores_Law.html.

[121] It is also common to cite Moore's law to refer to the rapid continuing advances in computing power per dollar cost. A similar progression has held for hard disk storage available per dollar cost. Another version claims that RAM (Random Access Memory) storage capacity increases at the same rate as processing power. However, it has been shown that software do not follow this same improvement pace. Wikipedia, http://en.wikipedia.org/wiki/Moore's_law

[122] Source code is code that a programmer can read and write. Object code can only be read by a computer and only consist of zeros and ones (e.g. 111000011000111). Meaning that one can not exclusively look at the object code in order to establish infringement, because those codes are not easily comprehended by humans and would need an computer to translate it. If one only has access to the object code one can establish the functionality of the program, but functionality is not protected by copyright.

[123] European Commission, Internal market, Frequently asked questions, Ibid.

[124] Ibid.

[125] 35 U.S.C. 112 and Article 83 EPC.

[126] Patents & Technology, Software patents: what one-click buy and safe air travel have in common, 2002 Duke L. & Tech. Rev. 0013, http://www.law.duke.edu/journals/dltr/articles/2002dltr0013.html

[127] Ibid.

[128] Microsoft, List of Bugs Fixed in Windows 2000 Service Pack 3, 14 Oct. 2004, http://support.microsoft.com/default.aspx?scid=kb;EN-US;320853 (In the list there are over a thousand bugs fixed in the 3rd service pack, which means that the bugs from the three previous versions are not included in this list http://support.microsoft.com/?scid=http%3a%2f%2fsupport.microsoft.com%2fsupport%2fservicepacks%2fwindows%2f2000%2fsp3fixlist.asp).

[129] Greeg Keiser, New Windows Bugs "Critical," Lack Patches, Information Week, Dec. 27, 2004, http://www.informationweek.com/story/showArticle.jhtml?articleID=56200518

The only way of debugging software is to change the source code, and the best way of detecting bugs is to release it to the public and get user feedback. Therefore, combining reusable program codes (called components) with few errors is the best way of making good software. The programmer weaves together thousands of ideas, where some of the ideas are new, but most of them are old ideas. "A patentholder could through his monopoly force licensees to use his tested, higher quality codes rather than to re-write them."[130] However, first one can not rewrite those since patent protects the idea no matter how the source codes are made. Second, one could turn it and say that patent and trade secrets[131] hamper the development of high quality software, in a way where *only* the owner of the patent or licensees of that patent can debug the product. This would lock the software development where big companies have patents on several key components, and can develop those without any real competition. If the debugging cost is too high or there is a little risk for the consumer to discover all of the bugs by normal use, big companies do not tend to do anything about it.[132]

Prior existing software is the base of good debugged software and they need to be in order to make them more reliable, because "higher quality means safer software".[133] Which means that access to source codes is the key of making safer software. This is especially important for software in aeroplanes, controlling medical devices and nuclear power plants which means that patenting software affect us all even if not using Internet or computers ourselves.

2.3. Why neither the proposal nor the EP's amendment fulfil the purpose

One of the aims with the directive is to "achieve the right balance between making patents available where appropriate in order to reward and encourage innovation, while avoiding stifling competition and open source development."[134] Therefore it is important to see what the proposed directive would allow in practise. My findings are that neither the original proposal, the amendments by the European Parliament, nor the Council's changes actually fulfil the aim of making the balance so that innovations will be both promoted and protected. My conclusion would be that the directive would clarify for patent offices in Europe to adopt the same broad standard that exist in America today, the same standard that European stakeholders are so afraid of and claim to try to avoid. Below are some examples.

The report by Arlene McCarthy has several amendments trying to clearly state that business-methods should not be patentable under the directive if they are non-technical,[135] this is to avoid fostering the development that has occurred in the US. At the same time, according to the report,

[130] Patents & Technology, Software patents, Ibid.

[131] **Trade secrets** are anything that includes any information not generally know "including a formula, pattern, compilation, program, device, method, technique or process" that has independent economical value and is subject to reasonable efforts to maintain secrecy. Uniform trade Secret Acts § 1 (4) of 1970 (for America), and Article 10 Bis Paris Convention of 1967 and Article 39 TRIPS of 1994 (for Europe).

[132] Following the **Cost-Benefit-Analysis**, a method that adds up the value of the benefits of taking a certain action, and subtract the costs associated with it. In order to see if the action is worth making. See e.g. Mind tools, Cost/Benefit Analysis - Evaluating Quantitatively Whether to Follow a Course of Action, http://www.mindtools.com/pages/article/newTED_08.htm; and Thayer Watkins, Introduction to Cost Benefit Analysis, http://www2.sjsu.edu/faculty/watkins/cba.htm. This is one of the reasons why punitive damages exist in America, to deter big companies from bad behaviour and force them to take actions in favour for the environment or consumers.

[133] Watts S. Humphrey, The Future of Software Engineering:1, news@sei interactive, 1st Q. 2001, at http://interactive.sei.cmu.edu/news@sei/columns/watts_news/2001/1q01/watts-news-1q01.htm; and Patents & Technology, Software patents, Ibid.

[134] European Commission, Internal market, Frequently asked questions, Ibid.

[135] E.g. Amendment 6 (Recital 13 a: new) and Amendment 17 (Article 4a: new) in Arlene McCarthy, Report on the proposal, Ibid. p. 8 and 14.

the status quo of the EPO's practise should not be changed.[136] However, the EPO already allow business methods that are patentable in the U.S. to be basically patented in Europe, because they are seen as processes fulfilling all the criteria for patentability.[137] Therefore, the EPO will be able to clearly follow the legal practise used in the US. America would not patent a pure business method without any technical features either, but is allowing a process with technical features to be patented.

Article 3[138] in the proposed directive leaves the door open for all software to be patented, since there would be presumption that the invention belongs to a field of technology. This means that the little difference the claimed difference "technical contribution" is supposed to make will be more or less eliminated, and the ability to grant patent as freely as in America would be in place.[139]

Article 4 (3) declare that technical contribution shall be determined by considering the state of the art and that patent claims should be regard as a whole, that can include both technical and non-technical features. This is basically what has been stated by the American court in *Graham v. John Deere*[140] where the obviousness test was refined into four steps,[141] but where the patent examiner still has to look at the subject matter as a whole.[142]

There are three things in the report from the Parliament that should get more attention; to impose the Commission to keep the sector under review, especially the impact on SMEs,[143] to urge the Commission to create a support network for SMEs in order to assist them in benefiting from the protection of intellectual property,[144] and to introduce a grace period.[145] These three measures would ensure that the SMEs would be able to take better part of the system and prevent big companies from abusing the system. Today, America is more or less the only country allowing a grace period of one year, which benefit innovators in a positive way.[146] Since software are essentially compilations of thousands ideas, it might not always be evident that a new and innovative idea has been created and without a grace period European innovator might miss their opportunity to patent. This would also create a better balance between European and American innovator, as well as foster more research and development (R&D) investments here in Europe.

[136] Amendment 10 (Recital 14) in Arlene McCarthy, Report on the proposal, Ibid. p. 10.

[137] Article 5 of the directive clearly state that computer-implemented inventions can be claimed as a process. The proposed Directive (COM(2002) 92), Ibid. Also in Amendment 9 (Recital 13d: new) in Arlene McCarthy, Report on the proposal, Ibid. p. 10.

[138] "Computer-implemented inventions as a field of technology. Member states shall ensure that a computer-implemented invention is considered to belong to a field of technology" Article 3, the proposed Directive (COM(2002) 92), Ibid.

[139] "By just using a computer" as stated in reports.

[140] 383 U.S. 1 (1966). The first obviousness test was developed *A. & P. Tea Co. v. Supermarket Corp*., 340 U.S. 147 (1950), i.e., "whether the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made."

[141] (1) Determine the scope and content of the prior art, (2) ascertain the differences between the prior art and the claims in issue, (3) resolve the level of ordinary skill in the pertinent art, and (4) evaluate any objective evidence of nonobviousness. Ibid.

[142] Is also further described in the USPTO latest examination guideline regarding computer-implemented business method inventions (last updated in November 2003). USPTO, Formulating and communicating rejections under 35 U.S.C. 103, Ibid.

[143] Amendment 21 (Article 7) in Arlene McCarthy, Report on the proposal, Ibid. p. 16 and 22.

[144] Arlene McCarthy, Report on the proposal, Ibid. p. 23.

[145] Amendment 24 (Article 8 e: new) in Arlene McCarthy, Report on the proposal, Ibid. p. 18.

[146] However, there are downsides with Grace Periods as well so more investigation on this would be needed before any implementation can take place, see e.g. http://www.patent.gov.uk/about/consultations/responses/grace/index.htm.

Part III. Resistance to patenting software - *patented software vs. free software*

Resistance against patenting software, the main arguments:

- Already sufficient protection through copyright
- Work against free trade and allows multi-national companies to block the market access
- Too expensive for SMEs, while it will strengthen and benefit big companies even more
- Will rise development expenses, increase the legal risk and insurance premiums

1. Free Software

1.1. Basic ideas of the Free Software Foundation[147]

The Free Software Foundation (FSF) was founded in 1985 by Richard Stallman. It is dedicated to promote computer users' right to use, study, copy, modify, and redistribute computer programs without any proprietary restrictions,[148] and is the principal organisational sponsor of the GNU Project.[149] Stallman's basic idea is to create a software free for anyone to use (LINUX), where the word free is equal to "freedom" and has nothing to do with the price.[150] The distribution and selling of their software is one of the biggest sources of funds.[151] To benefit from this free software the user agrees on never to patent or copyright any improvements he does, but to give away his improvements free to everyone to use.[152] Which follows the FSF's beliefs that restriction on software through patent hamper technical improvement and impede the good of the society.[153]

Normally, if a program is "free software" when it leaves the hands of its author, this does not necessarily mean it will be free software for everyone who has a copy of it. For example, public domain software (software that is not copyrighted) is free software; but anyone can make a proprietary modified version of it. Likewise, many free programs are copyrighted but distributed under simple permissive licenses, which allow proprietary modified versions.[154] In order to prevent the GNU free software from being turned into a proprietary one the FSF use a distribution term that they called "copyleft". Copyleft "uses copyright law, but flips it over to serve the opposite of its usual purpose: instead of a means of privatising software, it becomes a means of keeping software free."[155] To ensure the effectiveness of copyleft, modified versions must also be free and the user must provide the source codes of the modification. This ensures that work based on their programs becomes available to all users if it is published.[156]

[147] See e.g. http://gnu-tr.org/ (FSF´s homepage), and http://directory.fsf.org/ (co-operation between FSF and UNESCO).

[148] Free Software Foundation, http://www.gnu.org/fsf/fsf.html

[149] The name ``GNU'' was chosen because it is a recursive acronym for ``GNU's Not Unix''. GNU/Linux have almost the same features as Unix, that is one of the most common computer operating system, but Unix is proprietary restricted. FSF, Overview of the GNU Project, http://www.gnu.org/gnu/gnu-history.html

[150] Free Software is more of an ideology that values the freedom of creating software without restrictions. The name has lead to misunderstandings that the software would be given away free of costs, therefore the Free Software often use the expression "free as in free speech, not free beer" in order to explain it. FSF, Overview, Ibid.

[151] Ibid.

[152] The improvements must then also follow the GNU General Public License.

[153] Slackware Linux Essentials, Open Source and Free Software, at http://www.slackware.com/book/index.php?source=x68.html

[154] Richard Stallman, The GNU Project, http://www.gnu.org/gnu/thegnuproject.html

[155] The central idea of copyleft is that they give everyone permission to run, copy, modify and distribute modified versions of the program, but do not permit the user to add restrictions of their own. Thus, the crucial freedoms that define "free software" are guaranteed to everyone who has a copy; they become inalienable rights. Ibid.

[156] However, as long as the user only use his improvements for himself, he does not need to give it away. The specific implementation of copyleft that is used for most GNU software is the GNU General Public License, GNU GPL. They have other kinds of copyleft that are used in specific circumstances, for example GNU manuals are also copylefted, but use a much simpler kind of copyleft, because the complexity of the GNU GPL is not necessary for manuals. Ibid.

Mr Stallman started his career at Massachusetts Institute of Technology (MIT) in 1971 and worked in a group that exclusively used free software. At that time even computer companies often distributed free software and programmers were free to cooperate with each other, and often did. By the 1980s, most software was proprietary, which meant that it had owners who forbid and prevent cooperation by users. This was the reason for the creation of the GNU Project, a free operating system. Then Linux, a free kernel,[157] was developed by Linus Torvalds. Combining Linux with the almost-complete GNU system resulted in a complete operating system: a Linux-based GNU system.[158] Today there are estimates that hundreds of thousands of people use Linux-based GNU systems, including Slackware, Debian, Red Hat, and others.[159] A survey of 140 large North American companies made 2004 showed that 46% of them are using open software and that another 14% planned to do that soon.[160] This development has also been recognised by the European Union, where some sections of the institutions are in favour of the free/open source software in order to increase the competition on the software market with a positive outfall for the consumer.[161]

1.2. Open Source Movement

The Open Source Movement is another ideological movement within the Linux community that started through the Open Source Initiative in 1998. "Free software" and "open source" describe the same category of software and the same practical recommendations, that the source codes should be free/open, but they have different values.[162] The Open source prefers to base its arguments on the economic and technical merits of making source codes freely available to attract business users, rather than the moral and ethical principles that drives the FSF.[163] They argue that the open codes generate faster and better improvements of software than the "closed" once, notably the proprietary restricted software.[164] They do not offer a specific license, but support instead various types of available open source licenses. Their basic idea is to get more companies behind open source by allowing them to write their own open source licenses and have those licenses certified by the Open Source Initiative.[165] Because many companies want to release source code, but not necessarily use the GNU General Public License combined with the moral and ethical ideology that it comprise.[166]

Even if the Free Software and the Open Source are two different entities and disagree on the basic principles, their aim is almost the same and they can collaborate on specific projects and their "common enemy is proprietary software."[167] The Free Software clearly take a step away, not to be confused with the Open Source, because they believe that the OSI have opened up doors for businesses to misuse the term. Making users believe that the software is free to use and modify, but

[157] The essential part/the core of an operating system. Definition at Dictionary.com at http://dictionary.reference.com/search?q=kernel&r=67

[158] FSF, Overview, Ibid.

[159] Richard Stallman, The GNU Project, Ibid.

[160] Phillsbury Winthrop LLP, Demystifying common misconceptions about open source, patents and legal liability, 15 Dec. 2004, http://biz.yahoo.com/prnews/041215/nyw027_1.html?printer=1

[161] European Commission, DG Information Society, Free&Open software, 2 Feb. 2004, at http://europa.eu.int/information_society/activities/opensource/print_en.htm

[162] "Open source is a development methodology, free software is a social movement." Richard Stallman, Why Free Software is better than Open Source, http://www.gnu.org/philosophy/free-software-for-freedom.html

[163] Slackware Linux Essentials, Ibid.

[164] Open Source, http://www.opensource.org/

[165] With certain restriction on the licenses, for example the distribution must be free without discrimination, the source code must be available, they have to allow derivative work and respect the integrity of the original author of the source code. The Open Source definition, http://www.opensource.org/docs/definition.php

[166] Slackware Linux Essentials, Ibid.

[167] Richard Stallman, Why Free Software is better than Open Source, Ibid.

where the business is using the words "open source" to describe the product even if the software is in fact proprietary restricted.[168]

1.3. Software patents

> "Software patent are the software project
> equivalent of land mines: each design decision
> carries a risk of stepping on a patent,
> which can destroy your project"
>
> Richard Stallman
> Fighting Software Patents - Singly and Together[169]

The FSF do not seem to have anything against Intellectual property in general, or patent in particular, but only the fact of patenting software. Because every such patent covers some idea and the use of that idea, which by giving monopoly on patents inhibits the development of software.[170] One of FSF's reasons is because the U.S.PTO grants several thousands patents each year.[171] To read through all these and all the other that exist would be almost impossible. Pending patents are secret during the application time,[172] which can take several years. Further the formulation of the patent descriptions are so complicated that even the inventors do not recognise their inventions. Basically, Stallman had three approaches: (i) avoid the patent, (ii) license the patent, or (iii) overturn the patent in court.[173]

> i) which means not to use the idea that the patent covers, which can be difficult since programs are getting bigger and bigger. Some patents are very specific and a program can of course be without one or two special features, but it will be impossible to make good software when one can not include features that the users want. Other patents are too broad that they rule out an entire field, for example the idea of Public Key Encryption. By avoiding too many patents the program can become useless.[174]
>
> ii) to license a patent is normally quite expensive. If they ask for 5% of the gross net sale for using one patent, that can be affordable, but if one need to license 20 patents in order to make the software then it would be nearly impossible to make a profit. Stallman explain that there is one situation where licenses are very lucrative and that is for multinational mega-corporations, for example IBM with an extensive patent portfolio. The benefits of the collection of royalties for licensing their patents, is nothing compared to getting access to patents of others. Where they can threat to sue an independent inventor, unless he agrees to cross-license his patent because parts of his invention might touch upon one or several of their patents. To cross-license can be seen as an advantage, especially when compared to what a lawsuit would cost and that the independent inventor now is allow to use all those

[168] Ibid.

[169] Richard Stallman, Fighting Software Patents - Singly and Together, 2004, http://www.gnu.org/philosophy/fighting-software-patents.html

[170] "Compared with Copyright that covers copying and even if you have written a novel word-for-word as an already written novel, you can still defend yourself by proving that you got the idea and have never been exposed to the existing book. But with patent, even if you can prove that you had the idea on your own, that is irrelevant because it is covered by the monopoly anyway." Richard Stallman, Software Patents, Cambridge, UK, 2002-03-25, http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html

[171] During 2004, US PTO issued 187,170 patents. See *supra*.

[172] Most patent applications filed on or after November 29, 2000, will be published 18 months after the filing date of the application, or any earlier filing date relied upon under 35 U.S.C. 122, otherwise, all patent applications are maintained in the strictest confidence until the patent is issued or the application is published.

[173] Richard Stallman, Software Patents, Ibid.

[174] Ibid.

patents. However, the Mega-corporation will be able to use his patent and gets the right to compete with him.[175]

iii) the people granting the patents at the USPTO are not programmers themselves, which means that what is new, useful and non-obvious to them might not at all be new and non-obvious to programmers.[176] The criterion non-obvious is supposed to "be non obvious to someone skilled in the art," which basically means that for software to be patentable it should not be obvious to a programmer.

FSF has a clear picture of their view of patenting software, in a domain where they have clear expertise of programming. In a study of the economical impacts of patenting software written for the Commission the author writes, "The open-source community considers patents a threat to the development of open-source software and aims to ensure that patents do not affect such development. [...] However this position on patents could well change. Developers of open source software may find it advantageous to file patents to obtain bargaining positions e.g. licence money from owners of proprietary platforms." [177] By writing this, it shows that they did not entirely understand the basic idea of the free software foundation, that the FSF actually already make a profit of their software but that the idea is about the freedom of using, changing, copy, study and distribute the software without proprietary restriction. The already mentioned study also says that European independent software developers, just as developers of open-source software "are making disproportionately little use of the patenting possibilities open to them compared with the use made by large companies and by US SME and even independent software developers."[178] Contrary to the understandings of the writers the FSF has clearly understood what patent is, but their basic ideology withhold them from ever wanting to take advantage of the patent system.

[175] Ibid.

[176] Stallman gives several examples in his speech, but because it would make this paper too long I will not include all of them.

[177] R. Hart, P. Holmes & J. Reid, Study "The Economical Impact", Ibid. p.3.

[178] Ibid.

Conclusion - with special attention to SMEs and consumers

The American market indicates that one of the major risks that SMEs are facing is the disability to compete on the market. Trends are showing that big companies are either using their broad patents and demand a licensing fee from SMEs, in combination with a threat to sue them if they do not pay, or buy up the small companies. There are large companies with a considerable patent portfolio and a huge staff of lawyers just to enforce their rights. They normally tend to target SMEs, because they know that small companies do not have the funds to take the case further or to challenge the claimed patent. A Patent can be challenged if the abstract of the patent is too broad or ambiguous, but to challenge a patent is a very expensive and time-consuming procedure. This will make small companies to end up bankrupt or paying the license fee to the company. By having a lot of small companies paying the big company licensing fees, the big company will have "evidences" that their patent is valid, even if it is not, because why would so many pay licensing fees unless it is a valid patent? Further, the mentioned report to the Commission[179] also point out this risk and conclude that information to the SMEs would be enough. However, I do not see how information alone would make the patent system more and easier available for SMEs, nor cheaper.

IBM alone preliminary got about 3,248 patents during 2004, which gives a total of more than 28,500 U.S. patents in IBM's portfolio.[180] This would be IBM's twelfth consecutive year as the number one receiver of patents.[181] In January this year IBM pledge to open up access to 500 of their patents as a new way of managing their patent portfolio in order to let innovation benefit their customers, partners and society.[182] As well as "encourage and protect global innovation and operability through open standards." However, this pledge include less than 1.7 percent of their whole portfolio and IBM still has about 28,000 patents left.

A few other concerns that should be taken into account are the risks with too broad patents that might inhibit innovation, rather than promote it. There is a risk that the users will miss out on improvements of a software, because it is too close to an already patented invention those improvements will never reach the market. Also, in America there are companies whose business model is based on patent litigation as a threat and licensing as a revenue source.[183] They do not have any interest in selling a product or obtain customers. Therefore, more and more software companies offer their customers intellectual-property liability indemnification in case they would be sued.[184] Just showing how easily the patent system, that is supposed to promote and protect innovation, can be misused.

In 2002 the Federal Trade Commission (FTC) and the Justice Department in America held a joint hearing in order to discuss that the U.S. system regarding software patent has "run amok". Many corporations, including CISCO System Inc., and Intel Corp., testified that they are increasingly locked in intellectual-property arms race, stockpiling patents and spending millions of dollars each

[179] R. Hart, P. Holmes & J. Reid, Study "The Economical Impact", Ibid.

[180] USPTO's Releases Annual List of Top 10 Private Sector Patent Recipients for the 2004 Calendar Year, 11 Jan. 2005, http://www.uspto.gov/web/offices/com/speeches/05-03.htm; and IBM, IBM patents, http://www-306.ibm.com/software/data/awards/patents.html.

[181] During 2003 more than 40 percent of the 3,415 patents awarded were software patents. IBM Patents, Award archive (2004), http://www-306.ibm.com/software/data/awards/patents.html

[182] IBM, Feast for open source as IBM opens patent pantry, News, 2005-01-10, http://www.ibm.com/news/us/en/2005/01/patents.html

[183] Richard Wilder, How to fight against patent terrorism, CNET News, 6 Jan. 2005, http://news.com.com/2102-1014_3-5513518.html?tag=st.util.print

[184] Ibid.

year in patent lawsuits.[185] Lawsuits that has little to do with promoting or protecting innovation, and that software patents create a legal minefield where it is nearly impossible to write a program without using somebody's patent.[186]

If Europe does not allow patent of software or maintain status quo, the uncertainty of the legal situation will create a negative impact on the European market[187] and will maintain the big disadvantage when competing with companies outside Europe. But, Europe is about to blow their chance of creating a system that would take into account the failures already made in the US, by adopting the existing directive. Stepping right into a system that America has understood is not promoting innovation and has acknowledged as a bad way of dealing with software patents.

This is a complicated and a very technical area, and in this paper I have tried to present the important facts and information that has caused problems and misunderstandings. In order to wake up the debate and to try to find a solution that will work for all stakeholders.

By this paper I am not saying that patent is a bad thing, but I think that since America has already dealt with the issue and had problems with it, that we should learn from those mistakes and truly make a better system. But, how it looks today, **by adopting the existing proposal the main problems that America has been facing regarding software patents will not be avoided**. But, rather make the European industry to do the same mistakes. I propose that the Parliament's proposal to impose the Commission to keep the sector under review (especially the impact on SMEs), make the Commission to create a support network for SMEs in order to assist them to benefit from the system and to introduce a grace period should be implemented. However, this is not enough and I propose two principal solutions; clear practical guidelines in conjunction with a directive AND an easier accessible application systems.[188] None of these are addressed in the proposal.

By making clear, easier guidelines of how and when to patent a computer-implemented invention for the institutions granting patents would avoid granting broad and ambiguous patents. Because mostly it is the broad, ambiguous patents that inhibit the market the most and I would say inhibit innovations, while narrower patents actually could and would protect computer-implemented inventions. This is also important in order for the European market to be able to compete with the American enterprises, because if Europe would ban software patent this would give American companies a free ride of European inventions. Further, the application system of patenting should be made easier in order to reduce the cost of filing. Because, by lowering the costs it would encourage SMEs to use the patent system and this would in the end benefit consumers to get better, more and cheaper software. With this maybe the goal of giving all people access to ICT can become reality faster than if the European Union would allow the patent system to become as the one that has been so criticised in America.

The principle aim of the proposed Directive is to clarify the legal situation and make it more certain.[189] However this directive do not create any more legal certainty then there is today. The

[185] Andy Reinhardt, Inventing a Better Patent Law, BusinessWeek, 22 DEC 2003, http://www.keepmedia.com/ShowItemDetails.do?item_id=333764&oliID=255&bemID=I9RvNI5fRSugeR9Xr9wWQgaa7026

[186] Ibid.

[187] Hugo Lueders, Software patents ruling represent fair compromise, Financial Times, 7 Jan. 2005, http://news.ft.com/cms/s/810fb3c4-6052-11d9-bd2f-00000e2511c8.html

[188] This has been recommended to U.S. by the FTC, in order "to offer protection not suffocation." Andy Reinhardt, Inventing a Better Patent Law, Ibid.

[189] European Commission, Internal market, Frequently asked questions, Ibid.; European Parliament, The Legislative Observatory, Ibid.; European Commission, the Proposed Directive, Ibid. and Europa, Press Release, Patent: Commission welcomes Council agreement, Ibid.

proposal will not allow patents for any thing that can be patented today and it do not exclude anything, rather it does just follow the status quo without adding or removing anything. But, with one big difference, citizens and companies can go to the European Court of Justice to get clarity regarding the interpretation of the directive rather then to national courts.

The whole world is looking at how Europe will proceed in the matter and I suggest that we create a system that other parts of the world will follow. With a better, faster and smarter patent process, that will also reduce the cost of applying for patents. Hand in hand with a clear examination guideline for the patent offices to follow, where the requirements are specified in order to avoid broad and ambiguous patents. Then, finally the aim to set the right balance between rewarding and encouraging innovation[190] in Europe can come true.

[190] Ibid.