TRENTO LAW AND TECHNOLOGY RESEARCH GROUP
DIGITAL REPRINTS N. 4

lawtech trento

http://www.lawtech.jus.unitn.it

UNIVERSITY OF TRENTO
FACULTY OF LAW

LOOKING FOR A FEASIBLE FORM OF SOFTWARE PROTECTION: COPYRIGHT OR PATENT, IS THAT THE QUESTION?

Paolo Guarda

DECEMBER 2013 (ORIGINALLY PUBLISHED IN JULY 2013)

LOOKING FOR A FEASIBLE FORM OF SOFTWARE PROTECTION: COPYRIGHT OR PATENT, IS THAT THE QUESTION?

ABSTRACT

Software presents important aspects of heterogeneity compared to other intellectual works governed by Intellectual Property Rights (IPRs). This paper focuses on the analysis of the two main forms of software protection (copyright and patent), providing some considerations towards the proposal of a feasible system.

KEYWORDS

Intellectual Property Rights - Copyright – Patent – EU law - Sui Generis Right - Software

ABOUT THE AUTHOR

Paolo Guarda (email: paolo.guarda@unitn.it - Personal Web Page: http://www.lawtech.jus.unitn.it/index.php/people/paolo-guarda), PhD in Comparative Private Law, is Post-doc Researcher of Private and Comparative Private Law at the University of Trento (Italy) – Faculty of Law – The Trento Law and Technology Research Group. He teaches "Information Technologies Law" and "Intellectual Property Rights and Licenses" and is the author of several articles about issues related to Digital Age Law (Privacy, Copyright, Tecnology Transfer, ecc.).

Looking for a Feasible Form of Software Protection: Copyright or Patent, is That the Question?

SUMMARY: 1. Introduction – 2. Software and Copyright Protection – 3. Software and Patent Protection – 3.1. Premises and legal framework – 3.2. The comparative survey - 4. Conclusion

1. Introduction

Issues revolving around computer programs are an interesting field of research, since they represent the paradigmatic case of new products of human creativity made possible by the emergence of digital technologies. These have undoubtedly weakened the traditional legal set of rules that had hitherto governed creative products in the "paper era". Software possesses two natures[1]: the "literary" one, the source code written by the programmer using one or more programming languages that can, albeit with some difficulties, be associated with the "traditional" written text, and the "technological" one, which provides functionality and industrial application.

The software presents important aspects of heterogeneity compared to other intellectual works protected by Intellectual Property Rights (IPRs). Actually, it is a presentation of technical information, i.e. instructions given to a computer to perform its various tasks. Like other works protected by copyright, a computer

---

* This paper contains the post-print version of the refereed article "Looking for a Feasible Form of Software Protection: Copyright or Patent, is That the Question?" published in *European Intellectual Property Review*, Vol. 35, Issue 8, 2013, 445-454 (the content is identical to the content of the published paper, but without the final typesetting by the publisher).
I acknowledge my debt to the Max Planck Institute for Intellectual Property and Competition Law of Munich and the University of Trento (Technology Transfer Office) for the support received for my research. I would like to thank Prof. Roberto Caso, Prof. Laurent Manderieux, Prof. Giuseppe Bellantuono, Dr. Christian Heinze, Dr. Matteo Ferrari and Dr. Andrea Rossato for their valuable feedbacks, and Patrick McLoughlin and my longtime friend Eric Johnson, for the linguistic review. All errors remain my own.

[1] Actually the concept of "software" would cover much more than the simple "computer program", since it contains other further components (program description, accompanying material, etc.). However we will use both expressions as synonyms in this paper, since we are focusing on the phenomenon itself going into the semantic details and differences only when and if it will be deemed as necessary.

program can be represented by means intelligible to human language (so-called "source code") and is suitable to communicate (at least to an expert audience) ideas and information. In order to be used, this peculiar product needs to be decoded and understood by a machine (so-called "object code"). Unlike any other presentation of information, software is made available to users in ways (object code) which typically keep hidden the ideational and informative content.

The choice made initially by the international legislator, and then gradually by more local ones, was to protect this new form of creativity through the instrument which until then had managed other forms of "humanity's" creation: copyright. Debate regarding the benefits of copyright versus patents as the best form of protection has characterized years of doctrinal discussion[2].

What must always be kept in mind is that the chosen solution will have to be tested in the light of this simple but important assumption: preference should be given to the settlement that better ensures the promotion and dissemination of these new products of the technological development within the society[3].

Other forms of protection which may relate to software - often of particular importance in applicative scenarios - also deserve to be mentioned, although a deep analysis is beyond the scope of the paper. First of all we make reference to "trade secret", which refers to any confidential business information which provides a firm a competitive edge; the unauthorised use of such information by persons other than the holder is regarded as an unfair practice and a violation of the trade secret. Another important instrument can be represented by "trademark". This is generally composed of four basic elements: the name, the logo, the colors and the symbol (e.g. the apple of Apple, the horse of Ferrari or the Lacoste crocodile). The brand, therefore, is any sign capable of being represented graphically, particularly words (including personal names), designs, letters, numerals, sounds, shape of goods or the packaging thereof,

---

[2] *See* G. Ghidini & E. Arezzo, "Patent and Copyright Paradigms vis-à-vis Derivative Innovation: The Case of Computer Programs, 36 IIC 159 (2005) (describing how patent and copyright law address the case of derivative innovation in the software market and the likely consequences that the coexistence of the two paradigms would have on derivative innovations). *See* also S.J.H. Graham & D.C. Mowery, "Software Patents: Good News or Bad News?", in: R. Hahn (ed.), "Intellectual Property Rights in Frontier Industries: Software and Biotechnology" 45 (Aei Press, 2005).

[3] To begin with, an interesting analysis of the paradoxical evolution of software appropriation regimes is available in E. Harison, "Intellectual Property Rights, Innovation and Software Technologies. The Economics of Monopoly Rights and Knowledge Disclosure" 62-86 (Edward Elgar, Cheltenham, Uk – Northampton, MA (USA) 2008).

combinations or shades of color, as long as they are capable of distinguishing the goods or services of one firm from those of others. The importance of brands is linked to the traceability of a product or service at a given source and the reputation that is recognized. As part of the software distribution features, the ensemble of functionality, interfaces, security, architecture and performance help create the user experience that is then associated with a particular source or project. The relationship between brand and quality is reflected in the trademark licenses area. Finally, there are forms of technological protection that heavily affect the enforcement of statutory or private (*rectius*, contractual) rules. From this perspective we can mention the Digital Rights Management (DRM) systems: this terminology identifies the most advanced anti-access and anti-copy protection system available on the market, able to incorporate the copyright rules set by the copyright holder[4].

This paper focuses on the analysis of the two main forms of software protection: copyright and patent. From this perspective, the comparative study represents a pivotal methodology: European legislation will be the main point of reference for my research; when appropriate, details on the legislation of some European countries or of the U.S. model will be provided in order to compare different solutions. This will allow the main issues to be highlighted.

Following this introduction, the remainder of this essay is organized into two sections. The first section will be devoted to the description of the protection of software through copyright, and the second through patent system. Finally, in the concluding part some considerations will be developed towards the proposal of a feasible system of protection.

2. – Software and copyright protection

Although of interest in its own right, I touch only briefly on the history of the development of copyright legislation on software, and I generally do not tackle the description of (economic and based on personality rights) justifications that lead to the use of IPRs. I

[4] For an in-depth analysis, see R. Caso, "Digital Rights Management. Il commercio delle informazioni digitali tra contratto e diritto d'autore" (Cedam, Padova 2004) (digital reprint, Trento, 2006, available at the website: <http://eprints.biblio.unitn.it/archive/00001336/>); P. Samuelson, "DRM {and, or, vs.}the Law", 46 Comm. ACM 41 (2003); J.E. Cohen, "DRM and Privacy", 13 Berkeley Tech. L. J. 575 (2003).

instead focus primarily on the discipline in force in the European Union context[5].

The regulative framework here is basically composed of two legislative interventions: first, the Directive 91/250/ECC of 14 May 1991 on the legal protection of computer programs, then repealed by Directive 2009/24/EU of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs (Codified version)[6]; and second, on general copyright law, the 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[7].

The origin of the debate that preceded the introduction of the Computer Programs Directive of 1991 was due to the separation between hardware and software that had raised the issue of the appropriate form of protection for software that could easily be copied without the author's consent. Several solutions were made available: providing a framework specifically for this new product; protecting it through the patent system; considering it as a literary work and then protecting it under copyright law. When the debate on the form of protection for computer programs started in the 1970s, the position of the international community, within the "World Intellectual Property Organization" (WIPO), was to recognize a sui generis right, able to cover all the specific characteristics of this new product of human creativity[8]. WIPO published its "Model Provisions on the Protection of Computer Software" for supporting countries in amending legislation in the field of software starting from 1978. These considerations were summarized in a 1983 treaty proposal that has never been adopted. This model, indeed, was at the end not the choice of the national legislators; on the contrary, the idea of using the traditional copyright began to prevail[9]. The European Commission started deliberations in 1985, at the time when the "White paper on the Completing of

---

[5] As a point of reference for this part *see* C. Heinze, "Software als Schutzgegenstand des Europäischen Urheberrechts", 2 Jipitec (2011), available at: <http://www.jipitec.eu/issues/jipitec-2-2-2011/3082>.

[6] For further details *see* W. Blocher & M.M. Walter, "Computer Program Directive", in: M.M. Walter & S. von Lewinski (eds.), "European Copyright Law" 81-248 (OUP, Oxford 2010).

[7] For further details *see* S. von Lewinski, M.M. Walter, "Information Society Directive", in: Walter, von Lewinski (eds.), "European Copyright Law", supra note 6, at 921–1141.

[8] Regarding this part, see R.M. Hilty & C. Geiger, "Patenting Software? A Judicial and Socio-Economic Analysis", 36 IIC 615, 619-622 (2005).

[9] As we will see below in that period the U.S. legal system was moving to protect software by copyright, definitively codifying this rule in the "Computer Software Copyright Act" of 12 December 1980. From a certain point of view, the European solution could also be seen as a kind of legal transplant.

the Internal Market" was published. At that time, several States had already taken steps to explicitly recognize computer programs as copyrightable subject matter (first Germany, with the Copyright Amendment Act of 1985, and the United Kingdom, with the Copyright (Computer Software) Amendment Act of 1985)).

The solution based on copyright law prevailed both at international (art. 10(1) TRIPs; art. 4 World Copyright Treaty) and European (art. 1 (1) and recital 6 Dir. 2009/24/EU) levels. The regulation of software has special rules that differ from the general one (i.e. art. 1 (2) Dir. 2001/29/EC), which, however, remains always a "subsidiary scheme" for those elements that are not protected as computer programs.

The starting point is the art. 1 Dir. 2009/24/EC: member States shall protect computer programs by copyright, as literary works, within the meaning of the Berne Convention for the Protection of Literary and Artistic Works. Computer programs include their preparatory design material.

The protection provided for in this Directive applies to: i) the expression in any form of a computer program (ideas and principles which underlie a computer program or any element thereof are not included in this protection); ii) a computer program when it is original in the sense that it is the author's own intellectual creation; iii) computer programs created before 1 January 1993.

The main issues are the following: definition of computer program including preparatory design material; no protection of "ideas and principles"; and requirement of originality.

As to the first point, the Dir. 2009/24/EC does not offer any express definition; that is likely due to the fear of not immediately making obsolete the legislative choice. The meaning can be found in the case law of the Court of Justice of European Union (CJEU). The CJEU Case BSA (C-393/09)[10] set two main criteria: the object of the protection is the expression in any form of a computer program which permits reproduction in different computer languages, such as the source code and the object code[11]; the protection started from the moment when its reproduction would engender the reproduction of the computer program itself, thus enabling the computer to perform its task[12].

The protection is henceforth bound to the program code and to the functions that enable the computer to perform its task. This solution implies that there is no protection for elements without

[10] *Bezpecnostni Softwarova Asociace — Svaz Softwarove Ochrany v Ministerstvo Kultury* (C-393/09) [2011] E.C.D.R. 3 (BSA).
[11] *BSA* [2011] E.C.D.R. 3 at [35].
[12] *BSA* [2011] E.C.D.R. 3 at [38].

such functions (i.e. Graphical User Interface (GUI), or "mere data") and which are not reflected in the code (i.e. functionality in itself is not protected, since there could be a different code that may be able to produce the same function)[13].

As cited, the copyright protection covers also the preparative design material: this mentions a "structure" or a "flow chart". These materials should be capable of leading to the reproduction or the subsequent creation of such a program, respectively[14].

The second main point is represented by the famous idea/expression dichotomy: since copyright covers only the expression modality of an intellectual work, "*ideas and principles which underline any element of a computer program, including those which underlie its interfaces, are not protected by copyright under this Directive*" (see art. 1 (2) 2 Dir. 2009/24/EC). Only source code and object code of a computer program are forms of expression entitled to be protected by copyright. Other elements are not; for instance, the GUI does not enable the reproduction of the computer program, but merely constitutes one element of that program by means of which users make use of the features of that program. This apparently simple and easy to understand assumption gives rise to a whole array of problems. We refer, for example, to its application to program logic, algorithms and programming languages that is far from being devoid of criticality[15].

The third, and perhaps most problematic, element to be analyzed is that of originality. Art. 1 (3) of Dir. 2009/24/EC states that: "*A computer program shall be protected if it is original in the sense that it is the author's own intellectual creation. No other criteria shall be applied to determine its eligibility for protection*". Nonetheless, there are at least two limitations, as state in *BSA*[16]: "*where the expression of those components is dictated by their technical function, the criterion of originality is not met, since the different methods of implementing an idea are so limited that the idea and the expression become indissociable*"; and an author must "*express his creativity*

---

[13] *SAS Institute Inc v World Programming Ltd* (C-406/10) [2012] 3 C.M.L.R. 4,which explicitly decided this point.

[14] *BSA* [2011] E.C.D.R. 3 at [37].

[15] Recital 11 of the Directive is clear on the point: "*For the avoidance of doubt, it has to be made clear that only the expression of a computer program is protected and that ideas and principles which underline any element of a program, including those which underlie its interfaces, are not protected by copyright under this Directive. In accordance with this principle of copyright, to the extent that logic, algorithms and programming languages comprise ideas and principles, those ideas and principles are not protected under this Directive*". This assumption is also confirmed in Case C-406/10, at. 39: "(…) *it must be stated that, neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of its functions constitute a form of expression of that program for the purposes of Article 1(2) of Directive 91/250*". *See* Blocher & Walter, supra note 6, at 102-105.

[16] *BSA* [2011] E.C.D.R. 3 at [37].

in an original manner and achieve a result which is an intellectual creation of that author". The originality requirements could create several problems, since the understanding of this precondition to copyright protection was not uniform in all European legislations at the time when the discussion began. The text of art. 1 (3) means "originality" in the sense that it is the author's own intellectual creation, with no other criteria being applied to determine its eligibility for protection[17]. The requirement is therefore addressed if the program is not copied from another[18].

As regarding to the issue related to ownership of software, the general practice is stated by art. 2: the author of a computer program is the natural person or group of natural persons who has created the program or, where the legislation of the Member State permits, a legal person; if several persons participate in the creation of a program, the exclusive rights shall be held jointly by these persons. In the event that an employee creates a computer program following the instructions given by his employer, the employer shall have exclusive rights to that computer program.

The holder of the rights on a computer program may do or authorise the following activities: the permanent or temporary reproduction of the program, or a part thereof; the translation, adaptation, arrangement and any other alteration of the program; the distribution of the program (art. 4). The legal tool used for this authorisation is termed "license". The Directive establishes another crucial restriction, namely "principle of exhaustion": the distribution right is exhausted after the first sale of a copy of a program in the Community if such first sale has been made by the right holder or with his consent (art. 4 (2))[19].

---

[17] Recital 8 of the Directive emphasizes it: "*In respect of the criteria to be applied in determining whether or not a computer program is an original work, no tests as to the qualitative or aesthetic merits of the program should be applied*".

[18] For a detailed analysis of originality requirement *see* Blocher & Walter, supra note 6, at 93–98.

[19] Member States have differently answered the question of to what extent contractual restriction of the distribution right may prevent the effect of exhaustion. The issue is even more problematic if we take into account the case of online transmissions. On this point see the last decision of the CJEU: Case C-128/11 *UsedSoft GmbH vs. Oracle International Corp.*, where the Court has stated that the user license agreement that a consumer, who downloads the copy of a program, concludes with a company receiving, in return for payment of a fee, a right to use that copy for an unlimited period represents a sale (not a license); then second acquirer of that copy and any subsequent acquirer are "lawful acquirers" of it within the meaning of Article 5(1) of Directive 2009/24/EC. For further details see Y.H. Lee, "UsedSoft GmbH v. Oracle International Corp (Case C-128/11) – Sales of "Used" Software and the Principle of Exhaustion", 43 IIC 846 (2012).

Some limitations apply to these exclusive rights (art. 5). A person having the right to use the computer program may make a back-up copy in so far as it is necessary for that use. This person may also observe, study, or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program without the agreement of the right-holder. Furthermore, the authorisation of the right-holder is not required when reproduction of the code and translation of its form are indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided that the following conditions are met: those acts are performed by the licensee or another person having a right to use a copy of a program; the information necessary to achieve interoperability has not previously been readily available; those acts are confined to the parts of the original program which are necessary in order to achieve interoperability.

It is also possible to make recourse to general protection under copyright law. For instance, this is the case for subject matter not qualified as a computer program: GUI (see Case C-393/09); user manuals (Case C-406/10); the audiovisual component of computer games; and even programming languages and the format of data files (see Case C-406/10). From this perspective, various parts of a work will be protected under art. 2 (a) of Dir. 2001/29/EC, insofar as they contain some of the elements which are the expression of the intellectual creation of the author of the work (see Case C-5/08)[20].

I devote this final part to a concise description of the copyright on software in the United States of America[21]. Some of the rules previously expressed are shared by this legal system. Below are some brief details with regard to the U.S. without, of course, any claim to completeness. Computer programs are protected as literary works under the definition stated in the Copyright Act, § 101, Title 17 U.S.C (1976)[22]. After the approval by the Congress of the new

---

[20] *See Infopaq International A/S v Danske Dagblades Forening* (C-5/08) [2009] E.C.R. I-6569; [2012] Bus. L.R. 102. In order to complete the subject *see* a paper arguing against copyright protection for the non-literal elements of computer software: C. Heer, "The Case Against Copyright Protection of Non-literal Elements of Computer Software", 18 I.P.J. 1 (2004/05).

[21] For further information of a general nature on this issue, see the following U.S. reference works: M.A. Lemley, P.S. Menell & R.P. Merges, "Software and Internet Law", 4th ed. (Aspen Law & Business, New York 2011); R.P. Merges, P.S. Menell, M.A. Lemley, "Intellectual Property in the New Technological Age", 5th ed. (Aspen Publishers, New York 2010).

[22] Before the approval of the Copyright Act of 1976 the classification of the software in the context of works protected by copyright presented some problematic profiles. In the first interpretation, based on the wording of Title 17 of the U.S. Code, as derived by the Copyright Act of 1909, a work that could be seen only with the aid of a machine could not be protected by copyright: *see*

rules, a special commission ("Commission on New Technological Uses of Copyrighted Works (CONTU)) was launched in order to investigate aspects related to the relationship between copyright and technological progress. The conclusions of this work were endorsed by the Computer Software Copyright Act of 1980: it amended the discipline of 1976 and introduced rules specifically designed for the protection of computer programs.

Copyright for computer programs prohibits not only literal copying, but also copying of "nonliteral elements", such as the program's structure, sequence and organization. These non-literal features can be protected only "*to the extent that they incorporate authorship in programmer's expression of original ideas, as distinguished from the ideas themselves*"[23]. The graphics, sounds, and appearance of a computer program may also be protected as an audio-visual work; thus, copyright can be infringed, even if no code was copied[24]. The set of operations available through the interface is not copyrightable in the United States under *Lotus v. Borland*[25], but it can be protected with a utility patent. In a case destined to make the history of U.S. copyright on software - *Apple v. Microsoft*[26] - but with significant effects also on a global level, the court established that the look and feel copyright claim must demonstrate that specific elements of a user interface infringe on another work; then, a program's particular combination of user interface elements is not copyrightable[27].

Several aspects may be examined with respect to the protection of copyright on software in case law. An issue that certainly cannot be forgotten here is the one related to the idea-expression dichotomy: that happens especially when the defendant has done something more than copy the source or object code

---

White-Smith Music Pusblishing Company v. Apollo Company, 209 U.S. 1 (1908). For further details *see* A. Rossato, "Diritto e architettura nello spazio digitale – Il ruolo del software libero", 130-139 (Cedam, Padova, 2006).

[23] See *Computer Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992), where the Abstraction-Filtration-Comparison test for identifying these protected elements has been proposed. This test is aimed at distinguishing copyrightable aspects of a program from the purely utilitarian and the public domain.

[24] See *Stern Elecs., Inc. v. Kaufman*, 669 F.2d 852, 855 (2d Cir.1982).

[25] *Lotus v Borland* 516 U.S. 233 (1996).

[26] *Apple v Microsoft* 35 F.3d 1435 (9th Cir. 1994).

[27] With regards to the so called "look and feel" test see *Whelan Associates Inc. v. Jaslow Dental Laboratory, Inc., et al*, 230 USPQ 481 (3rd Circuit 1986), dealing with the question whether, even if there was no copying of object code or source code, there could be copyright infringement in copying the "overall structure" of a program; *Johnson Controls v. Phoenix Control System Inc*, 12 U.S.P.Q. 2d 1566 (9th Cir 1989), focusing on the question of whether the structure, sequence and organization of the alleged infringement was the same as that of the plaintiff's program; *Lotus Dve Copr v. Paperback Software*, 23 U.S.P.Q. 2d 1241 (1992), taking into account the flow charts of the parties to determine whether there was infringement.

expression of the copyright owner's program. The courts tended to favor the copyright owner. In one case, instead, the argument was successful: *The Q-Co Industries, Inc. v. Hoffman*[28]. The court denied a preliminary injunction to prevent the defendants from selling a program that was inspired by the program of the copyright owner in a case where the alleged infringer had taken a program designed to run on one type of hardware and had expended substantial effort to produce a program that would fulfill the same function on different hardware. Although the general structures of the two programs were very similar, they were written in different languages and employed completely distinct algorithms. Then, the court stated that the similarities "*can be more closely analogize to the concept of wheels for the car rather than the intricacies of a particular suspension system (…) it was the idea that was used rather than its expression. Therefore, copyright infringement has not been established*".

A peculiar aspect deserves to be emphasized: despite what happens in the context of continental copyright, the works that have not been fixed in any tangible medium of expression are not covered by the copyright protection (fixation principle).

In the end, regarding the protection time, for works created after 1 January 1978, copyright, in general, lasts for a period of seventy years from the death of the author, or in the case of joint works from the death of the last of the authors of the work. In the case of works made for hire, as we will analyze below, protection runs for a period of ninety five years from publication or one hundred and twenty years from its creation, whichever comes first[29].

3 – Software and patent protection

3.1 – Premises and legal framework

The current European scenario in terms of patent protection of inventions implemented by computer programs is the result of a complex evolution influenced by many factors[30].

---

[28] *Q-Co Industries, Inc v Hoffman* 228 U.S.P.Q. 554 (SDNY 1985).

[29] Copyright Act, 17 U.S.C. §302.

[30] Starting from a broader perspective, the patent system involves several, different treaties and conventions. For a general reconstruction of the regulatory framework, see D. Closa, A. Gardiner, F. Giemsa & J. Machek, "Patent Law for Computer Scientists. Steps to Protect Computer-Implemented Inventions" 4-8 (Springer, Heidelberg 2010). N. HOPPEN, "Software Innovations and Patents. A Simulation Approach" 16-20 (ibidem, Stuttgart, 2005). As regarding to the last proposal (13 April 2011) by the European Commission of a "Unitary Patent Package" (then positively voted by the European Parliament on 13 December 2012), and consisting of two acts: a Regulation on the European patent with unitary effect (Proposal of Regulation of the European Parliament and of the Council implementing enhanced cooperation in the area of the creation of

We described before that, when the sui generis protection was proposed at the International level, the U.S. legal system was moving to protect software by copyright. This (new) approach was codified both at the European level, through the Directive 91/250/EEC (art. 1)[31], and at the International level, via the TRIPs Agreement (art. 10)[32], establishing that computer programs are protected as literary works in accordance with the Berne Convention.

At the same time, however, the U.S. case law began to adopt a more permissive approach with regard to software patentability, announcing an increase in the number of patents granted in the 1980s and 1990s. The new trend had an international echo in art. 27 (1) of the TRIPs Agreement, which states that "*patents shall be available for any inventions, whether products or processes, in all fields of technology*". At the European level it influenced the European Patent Office (EPO)[33] praxis, bringing it to adopt a less restrictive attitude in granting patents via the use of sophisticated legal interpretations[34].

---

unitary patent protection, COM(2011) 215 final of 13 April 2011, as amended by council doc. 17578 of 1 December 2011), and an Agreement on a Unified Patent Court for litigation on infringements and revocation of European and unitary patents (Draft agreement on a Unified Patent Court and draft Statute, Council doc. 14750/12 of 12 October 2012)) *see* R.M. Hilty, T. Jaeger, M. Lamping, H. Ulrich, "The Unitary Patent Package: Twelve Reasons for Concern", 2012, available at MPI Web site: <http://www.ip.mpg.de> (raising concerns about this last proposed Unitary Patent Package on three main headings concerning: the complexity of the regime, the imbalances in the system, and the lack of legal certainty for investments in innovation); see also M. Lamping, "Enhanced Cooperation – A Proper Approach to Market Integration in the Field of Unitary Patent Protection?", 42 IIC 879-925 (2012); T. Jaeger, "Back to Square One? – An Assessment of the Latest Proposals for a Patent and Court for the Internal Market and Possible Alternatives", 43 IIC 286-308 (2012).

[31] Art. 1, Directive 91/250/ECC: "*In accordance with the provisions of this Directive, Member States shall protect computer programs, by copyright, as literary works within the meaning of the Berne Convention for the Protection of Literary and Artistic Works*". As an implementation of these provisions within the Member States, see for example: § 2 (1)(1) *Urheberrechtsgesetz* (UrhG); art. 1 (2) e 2 (8), law 22 April 1941, n. 633 "Protezione del diritto d'autore e di altri diritti connessi al suo esercizio" (Italian Copyright Law).

[32] "Agreement on trade-related aspects of intellectual property rights" (TRIPs), 15 April 1994 in Marrakech (Morocco).

[33] The "European Patent Office" (EPO) plays a pivotal role, by granting European patents for the contracting States to the "European Patent Convention" (EPC), which was signed in Munich on 5 October 1973 and entered into force on 7 October 1977 (The official EPO Web site: <http://www.epo.org/>). The EPO represents the executive arm of the "European Patent Organization", which is an intergovernmental body set up under the EPC. It provides a legal framework for the granting of patents via a single, harmonized procedure: a single patent application may be filed at the EPO at Munich, at its branches at The Hague or Berlin, or at a national patent office of a contracting State, if the national law of the State permits so. Currently,

Recording this troublesome difference in terms of legal certainty between the rule established by statutory law and practical application, at the end of the last century the European Union tried to advance the adoption of a directive that sought to address the question with greater transparency and rigor. With the proposal of the Directive COM/2002/0092 of 20 February 2002, the European Commission set itself the ambitious goal of harmonizing legislation on patents on computer-implemented inventions, making more transparent the conditions of patentability of computer programs, with respect to the not so often clear "case law" solutions[35]. This proposal was strongly criticized and challenged, especially by the open source community[36]. The text first received numerous amendments that have resulted in a new draft of 24 September 2003. The troubled history of the proposal continued when, on 18 May

---

there are thirty-eight contracting States. This Convention is further analyzed below.

[34] On this issue Hilty & Geiger, supra note 8, at 620 write: "*It is probably the desire not to hinder European companies, in relation to their American competitors, that has motivated the European Patent Office progressively to adopt a less restrictive attitude*"; and further: "*the practice of the European Office is the cause of the legal uncertainty of the legislation of the member states must be adjusted to the existing practice*". *See* also P. Leith, *Software and Patents in Europe*, 9 (CUP, Cambridge 2007): "*Board of Appeal 3.5.1 of the European Patent Office – which was given the workload for the relevant classifications – clearly took the view that the role of a patent office was to give protection to technological developments and that since such developments were happening in software control of hardware, they should be protected*".

[35] Hilty & Geiger, supra note 8, at 625, dealing with the definition of "technical character" write: "*Nevertheless, at least the solution proposed in the directive had the advantage of being clear, which cannot be said for the jurisprudence on this point*". For further analysis *see*, *ex plurimis*, W. Tauchert, "Patent Protection for Computer Programs – Current Status and New Developments", 31 IIC 812 (2000); A. Laakonen & R. Whaite, "The Epo Leads the Way, but Where to?", 2001 EIPR 244.

[36] For an analysis of the proposal and of the arguments of proponents and detractors of software patentability *see* S. Bisi, "Gli aspetti giuridici della tutela brevettuale del software", 2005 Ciberspazio e dir. 423, 435-442. More specifically, the criticisms that the Open Source (OS) movement aimed at software patents can be based on the following assumptions: the inadequacy of patent instrument in a sector with high technological content where innovation is incremental; the possibility that an overarching technical standard patent is an obstacles to OS; the pursuit of purely economic interests made by large software companies through the so called "cross-licensing"; the possibility that the results emerging from the OS world come to be prey to illegitimate appropriation by patent applications: *see* M. Välimäki, "A Practical Approach to the Problem of Open Source and Software Patents", 2004 EIPR 523. On the OS phenomeno in general, *see* H.J. Meeker, "Open Source Alternative: Understanding Risks and Leveraging Opportunities" (John Wiley & Sons, Inc., Hoboken (New Jersey), 2008); L. Rosen, "Open Source Licensing. Software Freedom and Intellectual Property Law" (Prentice Hall, Saddle River (New Jersey), 2005). An interesting analysis of the emerging "Open Patenting" (OP) phenomenon within the boundaries of OS can be found in M. Maggiolino & M. Lillà Montagnani, "From Open Source Software to Open Patenting – What's New in the Realm of Openness?", 42 IIC 804-824 (2011).

2004, the Council of Ministers decided not to take into account most of the work done by Parliament and to resume almost completely the original proposal. After a controversial debate on 6 July 2005, the European Parliament rejected, with a large majority, the proposal to adopt the directive on computer software, most likely determining the death of such a legislative initiative[37].

The following section will be devoted to providing a comparative investigation in order to determine the characteristic features of the patentability systems of computer-related inventions in some jurisdictions and institutions of reference (EPO and United States). When deemed appropriate, the discussion will be enriched through references to case law.

3.2 - The comparative survey

In this section I will give ample space to the discussion and description of the system operating in the EPO, since it is relevant for our analysis profiled primarily on the European legal scenario. Dealing with the implementation among the European countries of the Munich Convention, I will pinpoint the main problematic issues arising from possible differences in the application of the "European Patent Convention" (EPC) principles within the different jurisdictions. The study of the U.S. experience will be useful as it will provide the reference characteristics of a legal system that has influenced, and continues to strongly influence, rules and solutions on this peculiar issue[38].

Talking about software patentability within Europe, it is crucial to deal with the regulation set by the EPC and by the interpretations of its articles that EPO has offered[39]. Starting from the general requirements for patents, they are defined in art. 52 (1) of EPC*:* "*European patents shall be granted for any inventions, in all field of technology, provided that they are new, involve an inventive step and are susceptible of industrial application*". The most relevant paragraphs for our research of this provision are the second, point 3, and the third

---

[37] See Hilty & Geiger, supra note 8, at 621-622.

[38] For further details on the comparative survey *see*: Closa, Gardiner, Giemsa & Machek, supra note 25, at 20-30 (with some interesting references also to the Japanese patent system); M. Ranieli, "Cronache in tema di brevettabilità delle invenzioni software related con particolare riguardo al ruolo dell'EPO e alla più recente giurisprudenza del Regno Unito", 2009 Riv. dir. industriale, I, 233; D-H. Koo, "Information Technology and Law. Computer Programs and Intellectual Property Law in the US, Europe, Japan, Korea" (Pakyoungsa, Seoul 2005); N. Hoppen, "Software Innovations and Patents" 21-92 (ibidem, Stuttgart 2005).

[39] As main references for this part *see*: Ranieli, supra note 38, at 233-246; Hoppen, supra note 38, at 54-74; Closa, Gardiner, Giemsa, Machek, supra note 25, *passim*; K. Beresford, "Patenting Software Under the European Convention" (Sweet & Maxwell Limited, London, 2000).

one: "*The following in particular shall not be regarded as inventions within the meaning of paragraph 1: (…) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers*" and "*Paragraph 2 shall exclude the patentability of the subject-matter or activities referred to therein only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such*".

The exclusion from patent protection is not made, then, in terms of absoluteness, but is instead relative: it is limited to the case where the claims contained in the patent application relate to the categories excluded "as such". This prohibition leaves open the possibility of a more or less restrictive reconstruction of the requirements of patentability of inventions implemented by computer programs. Thus, the meaning given to the "technical character" of the software assumes considerable importance. Although this requirement is not expressly required by any provision, it is widely believed to be a common element characterizing the patentable inventions[40].

At the application level, it is obviously the EPO, when processing the examination, that has had the opportunity to deal with the interpretation and application of art. 52. What is quite interesting for our purpose is not only the decisional practices of the Office, but also the case law of the "Technical Board of Appeal" (TBA), internal office boards of appeal. We will provide a brief and quick outline of the more relevant ones[41].

The first argued application of article 52 EPC is recorded in the *Vicom* decision (1987)[42]. Here the Board finds evidence to discriminate the inventions consisting of mere computer programs from those which may instead receive patent protection on the so called "technical contribution", made by the invention to the state of the art. Thus, not all software should be considered excluded from patenting: the distinction is made based on the effect of the invention. The result of the invention must therefore be taken into account for assessing whether it achieves a transformation of a physical entity or is confined to expose a further mathematical value[43].

---

[40] See G. Paterson, "The European Patent System. The Law Practice of the European Patent Convention" 9-13 (Sweet & Maxwell, London 2001).

[41] For an *excursus* of EPO cases, at first glance, see Hoppen, supra note 33, at 66-74; Ranieli, supra note 33, at 243-246.

[42] T 208/84 (*Computer-related invention/VICOM*, OJ 1987, 14).

[43] This interpretation received an extensive (and in some respects exaggerated) reading. It was thus argued that the "technical contribution" could be satisfied with the claim of a physical entity "incorporating" the program: hence the praxis rule of patent applications of claiming, not the software as a product, but as a part of an apparatus or as a mechanism of a larger process. See also Hilty & Geiger, supra note 8, at 623-630.

This interpretation, however, has recently been overturned. Since 2000, there is in fact a new "case law" of TBA that reemphasizes the importance of assessing the contribution or technical effect of the invention, for the benefit of the application of discriminating criterion of "inventive step"[44]. This specifically happened in three main cases which have laid the foundation for a new interpretative approach called "the any hardware approach". In the *Pension Benefit* decision, the Board established that: "*There is no basis in the European Patent Convention for distinguishing between "new features" of an invention and features of that invention which are known from the prior art when examining whether the invention concerned may be considered to be an invention within the meaning of Article 52 EPC. This there is no basis for applying this so-called contribution approach for this purpose*"[45]. In the following *Hitachi/Auction Method* decision, the Board admitted that an application that includes any technical means has, for that reason, technical character and cannot be excluded from patenting. This interpretation was not deemed too extensive, and (then even potentially) harmful, since it was still necessary that the invention meets other decisive tests, for example, to be new, a solution not obvious to a technical problem, or capable of industrial application. The non-obviousness of the invention was summed up in the "inventive step"[46]. The last case that deserves mention is the *Microsoft/Data transfer with expanded clipboard formats*, where the TBA finally argues that: "*The computer program recorded on the medium is therefore not considered to be a computer program as such*"[47].

The practice of the EPO has therefore resulted in the granting of European patents on software itself, as considered and claimed; that is, actually, a recognition of the possibility of patenting software as a product[48].

The contracting States of EPC have implemented the contents through national laws which, in turn, have transferred to the domestic laws the ambiguous drafting of article 52 of the Convention. For instance, Italy has implemented the Munich Convention with the d.p.r. n. 338/1978; in particular article 12 of r.d. 29 June 1939, n. 1127 (today art. 45 legislative decree 10 February 2005, n. 30 "Codice della proprietà industriale (Italian

---

[44] Hilty & Geiger, supra note 8, at 628: *"(…) legal doctrine often suggests that this condition is the most appropriate to prevent an "inflation" of patents in the field of software*".

[45] T 931/95 (*Controlling pension benefits system/PBS PARTNERSHIP*).

[46] T 258/03 (*HitachiAuction Method*). On the risks of too broad an interpretation the Board says: "*the requirement for the claims of a patent application to relate to an 'invention' having technical character is just one test which must be passed to obtain grant of a valid patent. The invention must also be new, represent a non-obvious technical solution to a technical problem and be susceptible of industrial application*".

[47] T 424/03 (*Microsoft/Data transfer with expanded clipboard formats).*

[48] Ranieli, supra note 38, at 246.

Industrial Property Act)) has been modified in order to adapt the content to art. 52 EPC. The Italian legislator has kept unchanged the text of this latter article in order to avoid the risk of different national interpretations from those that may be credited to the European level. Even if Germany has developed its own practice on software patents (although mainly in line with that adopted by the EPO), we find in this legal system almost the same approach to the implementation of European legislation[49]. The topic of patentability of software is defined in § 1 of the *Patentgesetz* (as amended by the Law of 31 July 2009): "*Patents shall be granted for inventions in any technical field if they are novel, involve an inventive step and are susceptible of industrial application*". The second paragraph is dedicated to exclusion: "*In particular, the following shall not be regarded as inventions within the terms of subsection (1): 1. discoveries, scientific theories and mathematical methods; 2. aesthetic creations; 3. schemes, rules and methods for performing mental acts, playing games or doing business as well as programs for computers; 4. presentation of information*". Actually this exclusion works "*only when protection is sought for said subject matters or activities as such*" (§ 1 (3)). The interpretation of the term "as such" has obviously raised many problems. In a decision of the federal patent court "*Bundespatentgericht*" (BpatG) "software as such" has been defined as a program that is "*not directly associated with its functional execution*" in connection with hardware[50]. Analyzing the several patent requirements, the most relevant one appears to be the "industrial application". This is what, with respect to software, becomes the previously mentioned "technical character". In an attempt to elicit a proper explanation, the German Federal Court of Justice (BGH) stated: "*(…) Patent protection is available for a method describing a methodical action using of controllable physical forces to attain a visibly obvious effect obvious success without interposition of human intellectual activity*"[51].

The risk that uneven interpretations and applications become established in national law is particularly felt[52]. First of all, a difficult situation could arise if the European patent, in the case of a dispute, were to be submitted to a court of a designated member State. Since the national court is not bound to comply with the interpretation of EPO, it could consider the patent void, since it lacks the requirements for patentability or because the case falls among unpatentable situations. Another problem could be represented by the case in which a computer program inventor, rather than applying

---

[49] For an overview of German cases, *see* F. Rummler, "Computer Program Inventions Before the German Courts – A Review", 36 IIC 225 (2005); Hoppen, supra note 33, at 38-54 (for a studying between 1977 and 2003).

[50] BatG GRUR 1989, 42 – *Rollandensteuerung.*

[51] BGH GRUR 1969, 672 – *Rote Taube. See* Hoppen, supra note 33, at 24.

[52] *See* as reference paper for this part Ranieli, supra note 38, at 247-250.

for a European patent, directly asks for one or more national patents in the countries participating in the Convention. In this case, each patent office would grant or deny the request based on its interpretation of the prohibition in art. 52 EPC, leading to the possibility that the same patent application could have negative or positive results in various offices. Finally, were the inventor to assess the costs and risks of choosing where to apply for a patent and accordingly be orientated towards the EPO or to the more "complacent" national offices, the negative effects of a situation of objective legal uncertainty would be transferred to him.

This is the European legal scenario. We are now going to outline the main features of the U.S. patent system with respect to software patentability[53]. From a general point of view, the "United States Patent and Trademark Office" (PTO) has the power to process the patent filing and to manage the procedure for granting the protection. The issue is regulated first of all by Title 35 of the United States Code. The requirements for patenting are established by §§ 100-105. In particular § 101 states: "*Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title*". The criteria for granting a patent can be summarized as follows: the person is the inventor (§ 102); the invention is the proper subject-matter for a patent: machines, articles of manufacture, composition of matter, and processes (§ 101); the invention is "useful" and new" (§ 102) and "unobvious" (§ 103). It is worth mentioning that there are no explicit prohibitions like those in the EPC system (and then, generally speaking, in the European Countries)[54]. Regarding software, at the beginning of its history, in the 1970s, the dominant

---

[53] As reference for the U.S. analysis *see* A.K. Rai, J.R. Allison & B.N. Sampat, "University Software Ownership and Litigation: A First Examination", 87 North Carolina L. Rev. 1519 (2009), in part. 1526-1529; Hoppen, supra note 38, at 74-91. The debate over the extent to which software patents are likely to foster innovation is quite relevant in the U.S.: *see* J. Bessen & R. Hunt, "An Empirical Look at Software Patents", 16 J. Econ. & Mgmt. Strategy 157 (2007) (where the authors argue that software patents are substitutes for research and developments); R. Mann, "Do Patentes Facilitate Financing in the Software Industry", 82 Tex. L. Rev. 961 (2005) (where the commentator argues that software patents may push certain kinds of small software houses to attract financing).

[54] The statutory subject matter has always been considered as to "*include anything under the sun that is made by man*": this expression has been used originally by the Committee Reports accompanying the 1952 act (see S. Rep. No. 1979, 82d Cong., 2d Sess., at 5 (1952); H. R. Rep. No. 1923, 82d Cong., 2d Sess., at 6 (1952). Since then the Supreme Court has often used it with respect to any dispute regarding the patentability of certain dubious products: see *Diamond v. Chakrabarty*, 447 U.S. 303 (1980).

intellectual property regime was copyright[55]. In this regard, we mention a famous case, dealing with a computerized method for converting decimal numbers to binary numbers, in which the Supreme Court seemed to reject software as patentable subject matter on the grounds that patent law did not encompass abstract scientific or mathematical principles: *Gottschalk v. Benson*[56]. In the amendments themselves to the Copyright Act of 1976 of several years later, Congress expressly endorsed copyright as an appropriate protection regime for software. The interpretative turn occurred in the 1980s. In *Diamond v. Diehr*, the Supreme Court provided its first, clear indication of the fact that certain types of software-implemented inventions could be patentable[57]. The case dealt with the possible patentability of a rubber-curing process that used software to calculate cure time. According to *Diehr*, mathematical formulas in the abstract are not eligible for patent protection, but if the invention as a whole meets the requirements of patentability, and thus involves "*transforming or reducing an article to a different state or thing*", then it is patent-eligible, even if it includes a software component. This test has been subsequently commonly used by other courts[58]. The message was clear: software could be patented, but it had to be claimed as "something else"[59]. The key turning point happened in 1994, due to *In re Alappat*[60]. The Federal Circuit eliminated any kind of limitation on patenting software by arguing that subject matter criteria could be met by claiming software as a new machine, when it has been loaded on a computing device. This decision gave way to a series of positions taken in accordance with the point. In 1996, the PTO issued software guidelines that broadly allowed software as patentable whether it was claimed as a machine or a process[61]. In 1998, the Federal Circuit rejected any special subject matter test for

---

[55] *See* P.S. Menell, "An Analysis of the Scope of Copyright Protection for Application Programs", 41 Stan. L. Rev. 1045 (1989).

[56] *Gottschalk v Benson* 409 U.S. 63 (1972).

[57] *Diamond v Diehr* 450 U.S. 175 (1981).

[58] *See* for instance *Arrhythmia Research Tech, Inc. v. Corazonix Corp.*, 958 F.2d 1053 (Fed. Cir. 1992).

[59] *See* A.K. RAI, J.R. ALLISON, B.N. SAMPAT, *University Software Ownership and Litigation: A First Examination*, 87 *North Carolina L. Rev.* 1519, 1528 (2009) (available also at SSRN: <http://ssrn.com/abstract=996456 or http://dx.doi.org/10.2139/ssrn.996456>). Time by time copyright was becoming less attractive than the patent option and in the 1990s several appellate court decisions clarified that copyright was covering first of all the literal source code of the program: see *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F3d 807 (1st Cir. 1995), *aff'd*, 516 U.S. 233 (1996); *Computer Ass'n Int'l v. Altai, Inc.*, 982 F.2d 692 (2d Cir. 1992).

[60] *In re Alappat* 33 F.3d 1526 (Fed. Cir. 1994).

[61] *See Examination Guidelines for Computer-Related Inventions*, 61 Fed. Reg. 7, 478 (Feb. 28, 1996).

software in *State Street v. Signature Financial*, stating that software (like all inventions) is patentable if it produces a "useful" result[62]. Generally speaking, the approach to software patenting in the U.S. seems to be broad and lacking any real obstacles to interpretation.

Trying to pull the strings of this rapid comparative *excursus*, software is from a legal-dogmatic point of view excluded from the list of patentable inventions in the European context (see art. 52 (2) EPC), but broadly allowed in the U.S., where such an exclusion list does not exist in the statutory law. Despite these legal constraints, software patents have been and continue to be commonly granted in Europe, when, as seen above, a strong technical character could be demonstrated. It is clear how this additional legal discrepancy between the two sides of the Atlantic causes problems and costs that have a direct impact on investment strategies and, henceforth, on the possibility of future development and innovation.

4. Conclusion: pros, cons and real rationale of IPRs on software

I do not claim to provide any definitive answers with respect to the age-old question of the correct form of software protection. I wish to highlight some reference principles that should characterize, and influence, the legal regime in this particular field of application.

Scholars and commentators have extensively debated potential advantages and disadvantages of intellectual property rights in the software area[63]. As is well known, copyright prohibits slavish copy of a particular sequence of program lines, but does not preclude that the same idea be carried out by a different author through the use of other instructions. Therefore, the new program could be similar to the previous one, since the programming language is necessarily "poor", but it will still be protected if it is seen to be an expression of the intellectual work of the author. The important part of the software is its structure. What requires creative effort is not (or not always), in fact, the idea on which it is based, nor the function it will be called to perform. Rather, it is the particular order, the scheme and the economy with which instructions and cycles are expressed; that is, the form in which the programmer decides to express her ideas. The synergistic effect of combined protection of software as patent and as intellectual work covered by

---

[62] *State Street v. Signature Financial* 149 F.3d 1368 (Fed. Circ. 1998).

[63] *See* an overview of the literature on this issues and a list of pros and cons in K. Blind, J. Edler & M. Friedewald, "Software Patents. Economic Impacts and Policy Implications" 7-34, 167-170 (Edward Elgar, Cheltenham (UK) – Northampton, MA (USA), 2005). *See* also J. Weyand & H. Haase, "Patenting Computer Programs: New Challenges", 36 IIC 647 (2005).

copyright is still largely to be explored and certainly not a carrier of positive and simplistic legal regime.

Patent protection, instead, insists on the content of the work, on the creative idea, and takes the form of the exclusive right granted to the inventor in carrying out the invention and subsequent profit from it[64]. The historical origins of patents are intrinsically linked to the nature of protected inventions: products that for their achievement involve great expenditures of time, money and energy. This represents a first problematic issue with reference to software. Currently it does not need, in general, large investments and its realization usually depends on the work of some capable programmers for a certain period of time. Furthermore, the creation of an invention traditionally presupposes the existence of a verticalized entrepreneurial structure; as regards the software, instead, especially with reference to Open Source models and within the "Internet age", the production may be widespread.

Dealing with this kind of problematic issues, we have always to bear in mind that intellectual property rights are only justified if they are able to address the function they are intended for and their use and scope should therefore be analyzed in relation to their task. When this is no longer carried out by the legal tool, then a serious danger of over-protection immediately arises[65]. Intellectual property in itself represents a remedy to market failures inherent in a system characterized by the circulation of the information good[66]. In fact, it establishes a monopoly in the hands of the author/inventor capable of determining artificially those obstacles to the free use and access by others to their work that are not present in the state of nature. Like all monopolies, it still represents a second-best solution, accepted simply because it is considered the only way to encourage and foster innovation. Within this context, it should be understood that the instruments historically provided by the legal order are able to activate the virtuous circle which leads to new creative works and

---

[64] For an interesting social-economic analysis on software patentability *see* Hilty & Geiger, supra note 29, at 630-638. A discussion with much food for thought on the same issue is available in G. Floridia, *La brevettabilità del software in Italia e in Europa*, in *Dir. ind.*, 2004, 421 (arguing that software invention and its exclusion from patentability as such are not worth preventing the patenting of the content of computer programs). *See* also J. Cohen, M. Lemley, "Patent Scope and Innovation in the Software Industry", 89 Calif. L. Rev. 1 (2001) (arguing that software patents may also be conceived as a system to bypass the expressive form/idea dichotomy typical of copyright).

[65] *See* Hilty & Geiger, supra note 8, at 618; C. Geiger, "Fundamental Rights, a Safeguard for the Coherence of Intellectual Property Rights?", 35 IIC 270 (2004).

[66] *See* R. Caso & G. Pascuzzi (eds.), "I diritti sulle opere digitali" 21-22 (Cedam, Padova 2002), and the references listed there; G. Pascuzzi, "Il diritto dell'era digitale" 199-249 (Zanichelli, Bologna 2010). *See* also M.A. Lemley, "Property, Intellectual Propertym and Free Riding", 83 Texas L. Rev. 1031 (2004).

innovative ideas. The game is played fully on this aspect; software is a new matter on which to apply, in some respects, an old "legal arsenal".

From the point of view of patent, there are currently not enough empirical analyses regarding the impact on innovation of a wider software patentability. There are, however, a number of uncertainties and the current patent system has been complex, expensive and too difficult to be used and enforced for small entities[67]. It can safely be argued that the patent system is currently not perfectly shaped and adapted to the peculiarities of the software[68].

Indeed, any attempt to adapt the tools that the current intellectual property system provides will never be completely satisfactory and suitable to the needs to which this new product of human creativity gives rise. As some scholars and commentators have argued, we should move towards a new form of protection, "somewhere between" today's variously available protection systems. This idea is actually better than other more conservative versions which fail to understand that technological changes require a thorough reconstruction and recalibration of the "legal arsenal"[69]. This argument reflects the early vision of the WIPO, namely, the need for a sui generis right adapted to the present situation. What is needed is a system designed in such a way that represents and implements once and for all the basic assumptions on which the protection of proprietary ideas are based and, therefore, the claimed incentive of human creativity. Freedom is the prime mover in this context, both in terms of choices and in terms of content distribution for the benefit of all users, where these all, due to technological advances, more and more often become new creators themselves[70].

---

[67] Hilty & Geiger, supra note 8, at 631.

[68] In *id.*, 639-642, the authors propose some practical and judicial adjustments in patent law.

[69] *See*, *ex plurimis*, P. Menell, "The Challenge of Reforming Intellectual Property Protection for Computer Software", 94 Colum. L. Rev. 2644 (1994); P. Samuelson, R. Davis, D.K. Mitchell & J.H. Reichman, "A Manifesto Concerning the Legal Protection of Computer Programs", 94 Colum L. Rev. 2308 (1994); R.H. Stern, "A Sui Generis Utility Model for Protecting Software", 1 U. Balt. Intell. Prop. L. J. 108 (1993); E. Galbi, "Proposal for New Legislation to Protect Computer Programming", 17 J. Copr. Soc'y 280 (1970).

[70] Hilty & Geiger, supra note 8, at 646: "*In particular, it is necessary to remind oneself of the fact that in a liberal society, freedom is the principle, and exclusivity the exception. In other words, any extension of exclusivity, to the disadvantage of freedom, must be justified and explicable*".