

Remarks on the Patentability of Computer Software – History, Status, Developments

<http://swpat.ffii.org/events/2001/linuxtag/jh/index.en.html>

Jzef Halbersztadt

halber@timsi.com.pl

2003-09-14

An essay by Jozef Halbersztadt, patent examiner at the Polish Patent Office, prepared for a lecture in Stuttgart in July 2001. The essay expresses his private views on the needs to accommodate the peculiarities of software by specially tailored exclusion/reward rights rather than by just adapting copyright or, even worse, patent law. It narrates the history of software law as a chaotic evolution where the horses determined the direction of the cart until the cart ran against a wall and the driver woke up. Halbersztadt proposes a relaunch of the Samuelson-Kappor-Davis Manifesto of 1994 with various optional modifications and ways to introduce those ideas into the current European legislation process.

Contents

1 Why are these debates so difficult?	3
2 What is beyond patents?	3
3 What is special in software?	4
4 The first sui generis software law proposal 1965.	5
5 WIPO work 1974-1985	5
6 Software copyright 1980-1994	6
7 Chip Protection 1984-1989	6

8	Manifesto 1994	7
9	Software Act by Mark Paley 1996	8
10	Chip sui generis and software sui generis	8
11	Green Paper on the Patent System in Europe and other actions of the European Commission 1997-2001	9
12	What can be done	9
13	Table 1. Options of sui generis software law	10
14	Table 2 Amendments to the European Patent Convention (after November 2000)	11
15	References	12

This is an essay about a history of computer programs legal status and possible alternative to nowadays “status quo”. Computers have been with mankind more than 50 years. In fact they were there earlier. Fifty years ago it was already clear that they are important. One can say that mankind had enough time for any kind of preparations. I will try to describe them.

During the first decades of computer industry development, there was no problem with the software, hardware was the predominant source of value in the computer industry. Software was often “bundled” with hardware sales, given away with the hardware. Academic researchers created much of the early innovative software, particularly operating systems and utility programs. Software not being produced by hardware companies tended to be custom-developed under individual contracts. This situation has changed radically with the advent of the personal computer, which ushered in an era of companies devoted to producing and selling software packages.

Initially, software was delivered physically with a mainframe during the installation process. Given the high cost and relatively small number of transaction involved in buying a mainframe traditional contract and trade secrecy provided right legal framework. There was also no ambiguity in licensing: the software stayed with the physical machine, which was accessible to one user in one location.

Traditional contract law became impossible to use with the advent of mass-marketed, end-user software. The so-called shrink-wrap licenses arrived instead. They were with copyright protection, since copyright was seen as well-suited to protect information products that are publicly distributed to a mass market.

1 Why are these debates so difficult?

Debates about the legal status of computer programs were initiated before the massive growth in distribution became a fact. They were conducted in many places. With very modest results. What is striking is the failure to reach conclusions derived from the debates. Apparently the final decisions were born outside of the public debates. Instead of enlightened participation of interested circles in a decision-making process one can see a chaotic development under the constraints of inertia within the limits of existing laws.

It is easy to find the reasons. By making laws, the lawyers are influential. They hesitate very much to divide their trade into separate chapters and chapels. They used to advocate very strongly in favor of common law allowing only as the second best solutions extensions of traditional specialized laws, like patent law. Such laws, their institutions also are under firm control of traditionalists. Extensive using of traditional terms and claiming that they perfectly fit to the modern times is liked by politicians, who do not understand the merits, but like an illusion of understanding and controlling of developments.

Of course other professions have also their arguments. Although it is striking how marginal is an interest of economists to analyze an impact of various options. Still, sometimes non-lawyers, professionals of the trade that is under regulation do prevail. But it is not enough for real success. Non-lawyers implant to the main body of law does not mean final acceptance, because only lawyers and their close servants are mainly working with the law. Some parts of law used to be applied in legal practices some remain really dead.

Let's be specific. The most technology related law is a patent law. It is part of so called industrial property law. The second main part of the industrial property is a trademark law. It has nothing to do with engineering. Due to applications to computer software, copyright became more strongly related to technology as the trademark law. The patent law has really narrow limits. Their basic rules are very old, some were born in medieval times. The chapter law & technology does not offer much choice.

2 What is beyond patents?

For technical innovations some countries - Germany was probably the first - created simplified patents. They have various names, petite patents, industrial models, utility models. Their rules are much more diversified than patents. There are only a few international regulations devoted to them. As patents are often related to international trade and protection of investment abroad, serving to small enterprises that operate on local markets is usually the purpose of models and petite patents. Their definitions do not allow solutions from all fields of technology as it is the case with patentable inventions. Their popularity inspired an idea of special rules and protection certificates for particular fields of technology.

Distinctive laws for special (*sui generis*) fields of technology are very rare. Lawyer and patent lawyers usually claim that from their perspective all technology is the same

thing. This paradigm should be challenged and the challenge should start from software area. If any part of technology poses particular features it is software. Software is a new kind of entity, with the ability to transform all other technologies, including the creative arts, politics, and economics.

3 What is special in software?

What are distinctive features of software that are making software patents a nightmare and justify separate treatment?

- Take-up of digital products or protocols occurs so quickly that if the first to market enjoys patent protection, competing products can be easily excluded.
- Wide distribution and varied uses of digital products make patent infringements extremely difficult to detect, prove, and prosecute.
- Digital technology's ubiquity makes it nearly impossible for people using that a technology, directly or through the Internet, to determine whether they are infringing a patent or not, even having the best intentions.
- With ideas and software proliferating so swiftly and invisibly to so many recipients across the Internet, any patent infringement will likely spread quickly and widely among "small violators" whom it will be difficult, expensive, and unprofitable to pursue.

The special legal treatment for software should be seen as an alternative to its patentability on general terms. Paradoxically traditional thinking would deny patentability to software per se, as it has been denied to mathematical algorithms. It was a result of understanding that general knowledge like mathematics should not be privatized. Without decades of long separation from patent law the history of computer industry would have been quite different. If software patents were allowed in 80's to the extent they are accepted nowadays Microsoft would never won with IBM and became number one in software industry.

It should be put loudly and clearly. The problem of patents is a problem of monopolies. The protection does mean protection of monopolies. They are acceptable only if they are good for society. Intellectual property law is founded on securing the public good. The medieval idea behind patents was to give a privilege of limited in time monopoly to encourage innovations. The justification for granting a monopoly to an inventor or importer of a technology has long been that the grantee trains others to exploit the invention after the grant expires. Thus, the inventor or importer gets a short-term benefit from the monopoly, while the public gets a greater long-term benefit from the inventions later, unfettered use. Whether nowadays such profitable monopolies should persist is improper for lawyers or patent attorneys to judge. The answers should be left to the economists, politicians, technology specialist, in case of software to computing professionals. There are a very few such studies. And what economists have to say

is rather against providing monopolies for distinctive aspects of technology, especially software.

If the public gains no more benefit (and arguably far less) from the patent system in its current form than it would enjoy if no patent system existed, the system should be corrected. If the current situation benefits business concerns exclusively, either giant corporations that seek advantage over their competitors, or opportunistic corporations that acquire software patents solely to extract licensing fees from their competitors, it should not be left without response.

In the past there existed much reluctance with regard to patent protection of computer programs. In 1964 the US Copyright register had announced (1) its guidelines on the copyright registration of computer programs. Contrary to the other countries, particularly European, the US has the Copyright Office that issues certificates for register items. Nevertheless, relatively few computer program registrations occurred in the decade and a half after this policy recommendation.

4 The first sui generis software law proposal 1965.

The American initiative probably stimulated Helmut Oehlschlegel - a patent engineer of AEG company - to publish (2) a proposal of an international system for the protection of programs. The main objective of the proposal was to save duplication of work in developing programs and to use human resources more effectively. He suggested creating an international office that would register and deposit programs in object code together with a description of their practical use. In a periodical, information about new program registrations should be disseminated. With respect to the program utilization, the office should be functioning as the program store and the library as licensing agency. The user interested in a program should be able to rent a copy of that program by a remote librarian service against a renting fee. But they should be able to buy a copy of the program together with a license to use it. The renting and licensing fees should be transferred to the owner of the program who has to pay a certain administration fee to the international office.

It was proposed in West Berlin a quarter century before arrival of shareware and Internet. If we evaluate this proposal from a today's perspective it appears rather modern and fits to some special systems for software proposed only recently.

5 WIPO work 1974-1985

From the middle of 60's to the middle of 90's there were printed dozens of papers advocating special treatment of software in the US in academic and professional journals. Next to nobody had in mind software patents. Those who thought that existing law should apply to programs focused only on copyright. Opponents claimed that due to executable nature of programs copyright cannot cover their most characteristic and valuable aspects and therefore is insufficient. Outside the US consensus of professional opinion for special (sui generis) law was even stronger. The most official studies on special protection

for software were performed by WIPO. This work lasted from 1974 to 1985 and was initiated by the UN which asked WIPO to prepare a study on the appropriate form of the legal protection for programs and on measures to facilitate the access of developing countries to information on software. The request was based on the idea that a registration and deposit system could facilitate the dissemination of computer programs in favor of developing countries.

WIPO prepared twice a draft treaty to constitute an International Union for the Protection of Computer Software. The first of these drafts was presented (3) in 1976, also provided an optional and secret registration and deposit of the software to be protected. Such register was seen as an instrument of proof of the existence of a program at certain point of time, and in an accumulation of pool of freely available software once after the term of protection was terminated. The second draft treaty was presented (4) in 1983 no longer included such register. Both drafts were not accepted.

6 Software copyright 1980-1994

In time between WIPO produced a proposal that was not accepted as well but triggered to certain extent the preparedness to accept copyright protection for software. WIPO published (5) in 1978 the Model Provisions on the protection of Computer Programs which provide for a protection of the form of the expression and exclude concept, methods or algorithms. As the condition of the protection the model provisions required originality in the sense that the software must be the result of its creator's own intellectual effort. The protection right should grant to the proprietor the exclusive right to copy, disclose distribute and use the computer software, and also the right to derive from a program or from detailed program description another program. On the other hand the protection does not cover independently created software that turn out to be similar to software of another proprietor. The model provisions did not provide for any formalities as a condition of protections.

In pure sense the Model Provisions have never been implemented. They had however strong influence in defining the software elements for which protection was defined though copyright system. This system the US introduced (6) in 1980 and began a campaign to persuade the other countries to adopt the same approach. The first Department of State victory was in Japan, Europe followed (7) in 1991, and WTO/TRIPS of 1994 includes provision requiring the member states to protect computer programs by copyright law.

7 Chip Protection 1984-1989

In many respects software resembles semiconductor chips. In the US the Semiconductor Chip Protection Act (SCPA) was introduced (8) in 1984 after heavy lobbying by the semiconductor industry. After the passage of the SCPA, many industrialized countries quickly adopted their own laws protecting integrated circuit layouts.

Since the SCPA was the first sui generis integrated circuit protection law in the world the initial drafts of Treaty on Intellectual Property in Respect of Integrated Circuits

(“the IPIC Treaty”) were based on the SCPA. These drafts were objected by some developing countries that wanted the treaty to include compulsory licenses and a dispute resolution forum that was not controlled by the United States or other industrialized countries. In response, the IPIC Treaty was amended (9) to alleviate the concerns of the developing countries. Even though the treaty does not directly conflict with the SCPA - since compulsory licensing under the IPIC Treaty is not mandatory - because of these amendments the United States, and later Japan and European countries failed with ratification. Nevertheless all industrialized countries, and after TRIPS all WTO member countries protect topographies of semiconductor chips in similar way. In the US the SCPA provides automatic anti-cloning protection to semiconductor designs from the date of the first commercial distribution of a chip embodying them. This protection lasts for two years unless a chip developer registers the design. The registration process - like that for the copyright - involves only a light examination of the application. A timely registration will extend duration of protection to ten years.

Not only because hardware is interchangeable to software the special system for chips reflects many of the design principles that can be applied to software regime. There is however one important difference. Registration of software innovations would not be easy to achieve because there is no intermediate design document uniformly prepared by software developers that could serve as registration material like “mask work” for chips. Accustomed to secrecy developers of closed software would be reluctant to register a design document that disclosed all the internal design elements of their program, information that they can protect as trade secrets.

Therefore for the software it is appropriate a modified chip protection approach. A software developer might have opportunity to register only part of the program, for example, a new interface design, a macro language, a new algorithm, or the like, without having to register the product as a whole, as is required by chip protection law.

8 Manifesto 1994

They do not propose a substitution of copyright and general patent law for software in its totality but claim that there is a gap between both, which should be filled by sui generis type of protection for computer software. In this gap would fail many algorithms that are responsible for the behavior of a computer program and which represent incremental improvements. Such algorithms would usually not be patentable nor are protected by copyright.

What Samuelson/Davis/Kapor/Reichman proposed is a system, which provides for limited period of protection against cloning. They say that the period has to be long enough to give efficient incentives in the development of innovative software and short enough to prevent blocking of the competitors from entering the field. Supporters of Open Source would claim that the right period is zero. But revised Manifesto can embrace them offering for the code disclosure an additional privilege.

The original Manifesto proposes a registration system by which the programs during their anti-cloning protection period maybe registered and by which the programs are

being protected for the additional period of time during which they are available for everyone for taking a license on standard terms and conditions against payment of compensation. Apart from the registration, it is also similar to the operation of collecting agencies in the field of copyright as installed in number of countries in particular for works of music. The Open Source licenses can be integrated with the proposed system.

9 Software Act by Mark Paley 1996

The Software Act differs from the Manifesto primarily by far more clearly defining what is protectable. The Manifesto unnecessarily confuses protection by dividing software into five entities, each with different classes of protection: program code, whole program compilation, subcompilations, algorithms, and features. This problem is caused by the Manifesto's adoption of the narrow definition of the term "algorithm" from Webster's New World Dictionary of Computer Terms. The Software Act instead uses instead much broader definition of the term "algorithm" from the university-level computer science text *Pascalgorithms*. The broad definition may contain all five of the above software behavior entities, and protect them with a single consistent scheme. The Software Act allows the Patent Office to create different classes of royalty rates for different types of software behavior based upon its economic value. Regulatory classifications are more likely to be flexible enough to keep pace with the fast evolving field of computer science than a statutory distinction.

10 Chip sui generis and software sui generis

There are very few registered semiconductor topologies. In the US only one disputable case was investigated by court. In other countries the law was practiced even in lesser extent. Much more popular is patenting of hardware chip solutions. Therefore the sui generis law for chip topology is widely seen as a failure.

The double failure with sui generis system for computer hardware, the failure to ratify an international treaty and unimportance of internal laws, worked against special law for software. The 1990s are scene of widespread introduction of software provisions to copyright laws, first unilateral in the developed countries and later globally as result of TRIPS. In the US where software copyright was in force since 1980, limits of the copyright law to protect software were tested in details. In watershed decisions courts refused to extend protection on functional aspects of programs and some judges in writing directed plaintiffs to legislators or the patent office. Legislators were deaf, but patent offices turned out to be more helpful.

Argument that because sui generis for hardware was not successful we should not risk sui generis for software is wrong. The first situation means only that chip topology law turned out less attractive in relation to patents. The second situation is different. The main form of protection to software is copyright. Protection given by this law is too weak, at least for a part of the software industry. Such are conclusions from copyright cases in the US courts. Patenting software in the US and elsewhere is a result of a

failure to deliver the third intermediary system. Because of an inactivity of legislators everybody to whom copyright was not enough became potential client of the patent office. It is wrong in two aspects. Firstly, patents are giving much stronger and longer protection in relation this to given in any potential sui generis system, there is nothing like general compulsory license in the patent system. Secondly, because software solution should be described in specific “claim language” as a result we have not software patent but rather pseudo-software patent.

11 Green Paper on the Patent System in Europe and other actions of the European Commission 1997-2001

The Green Paper (12) and following actions are mainly devoted to introduction of unitary Community Patent, covering the entire territory of the European Union. Clarification of a protection of computer programs was there a marginal issue. The paper and those with whom it was consulted were very much for patenting the software. Till recently only one argument was used: it became possible in the US and this change had a very positive impact of the development of the software industry. EUROLINUX petition to the European Parliament resulted in public debates in some member countries. The drive to Americanize European patent law was checked, but position of DG Internal Market that is in charge of the issue in Brussels remains unchanged.

12 What can be done

Mark Paley’s Model Software Petite Patent Act needs only small corrections to accommodate to the world with free/open source software. Outline of such “face lifting” was proposed in option 1 of Table 1. The source of option 2 is an idea that it is not enough to introduce new legal status of software. Something should be done with rights granted in the past and applications from countries that will persist in allowing software patents. This conversion can lead to dangerous outcome. Because it assumes that it is possible to convert the conventional patent claims composed in a language used to human communication into new type of description chosen to describe behavioral aspects of a computer program. The option 3 is proposing more advanced than compulsory license mechanism of financial side of the system. Together these three options create a space where the most appropriate solution can be found.

Paley proposed his Software Act as an additional chapter in the patent law. This should be modified It would be reasonable to append the Software Act’s petite patent law onto the copyright regulations, where software currently receives its greatest protection anyway. The Software Act as a part of the copyright regulations should be compatible with Berne Convention. There is no need to conclude a new international treaty; it is enough to amend the Berne Convention. Because WTO/TRIPS Article 10.1 requires member countries protect computer software as “literary works”, thus granting with the Convention Article 7(1) lifetime-plus-50-years protection. Therefore to fully implement rules of the Software Act: short time of protection, international court etc.

there is a need to amend TRIPS. But many changes can be put into law without amending of TRIPS treaty. Since the Berne Convention does allow member countries to provide new protections outside the Convention's scope, which grant less than Convention's normal minimum rights, many options are open. A European Algorithm Office, European Algorithm Court on Paley lines can be established fairly quickly. There is also a need of accompanying measures. A more rational legal status for software requires some amendments in patent law. As the changes in copyright law some changes in patent law can be made without agreement with the US, without revision of TRIPS and some need such cooperation. Two versions of amending of the European Patent Convention are given in Table 2.

13 Table 1. Options of sui generis software law

	Manifesto 1994	Option 0 (Software Act 1996)	Option 1 (Add-on to Software Act)	Option 2
Object	One from number of alternative prototype frameworks ¹	Algorithms but with much broader definition of the term “algorithm” ²	As in Software Act Claims	Claims analogous to patents. ³
Action of developer	Registration with some disclosure. ⁴	Filling simple Copyright-like form. ⁵	As in Software Act plus a prior art statement by developer.	Formulations of claims
Action of public body	no patent-like examination process, later option to challenge.	Examination after the challenge has been made.	Some prior art search ⁶	invalidation incentives
Benefit offered to the title owner	A period of blockage or (choice by authors) automatic mandatory license on standard terms.	absolute exclusion ⁷	As in Software Act with differentiation between types of second comers ⁸	non-exclusive models ⁹

14 Table 2 Amendments to the European Patent Convention (after November 2000)

original (after 2000-11)	special exemption of software	reaffirmation of <i>technical invention</i> doctrine
Article 52 – Patentable Inventions ¹⁰	Create new paragraph 4 with the wording borrowed from current point (c) claiming that these inventions are not industrial applications. ¹¹	no change of law necessary but courts would have to return to the letter and spirit of the current law. ¹²
Article 53 – Exceptions to Patentability ¹³	Optionally create new point (d) with PROGRAMS FOR COMPUTERS transferred from current Article 52(2)(c) ¹⁴	–

15 References

- (1) Copyright Office of the USA, Copyright Registration of Computer Programs, 1964
- (2) Oehlschlegel, Sollen und koennen Rechenprogramme geschuetzt werden?, GRUR 1965, 465-468
- (3) Draft Agreement for the Protection of Computer Software and its International Deposit, WIPO Document AGCP/NGO/III/3, April 7, 1976
- (4) Draft Treaty for the Protection of Computer Software, WIPO Document LPSC/II/6, June 17, 1983
- (5) Model Provisions on the Protection of Computer Software, WIPO Publication No. 814, 1978
- (6) Computer Software Copyright Act, December 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (1980)
- (7) Council Directive of 14 May 1991 on the legal protection of computer programs, 91/250/EEC, 1991 OJEC. (L 122) 42
- (8) Semiconductor Chip Protection Act, November 8, 1984, Pub. L. No. 98-620, 98 Stat. 3347 (1984)
- (9) Treaty on Intellectual Property in Respect of Integrated Circuits, Washington, May 26, 1989, 28 I.L.M. (1989), 1477
- (10) Samuelson, Davis, Kapor, Reichman, A Manifesto Concerning the Legal Protection of Computer Programs, Columbia Law Review, vol. 94, No. 8, December 1994, 2308-2341¹⁵
- (11) Paley, A Model Software Petite Patent Act, 12 Santa Clara Computer & High Tech. Law Journal 301, August 1996¹⁶

¹⁵<http://wwwsecure.law.cornell.edu/commentary/intelpro/manifint.htm#intro>

¹⁶<http://hometown.aol.com/paleymark/ModelAct.htm>

(12) Green Paper on the Community Patent and Patent System in Europe, COM(97), 314 final, June 24, 1997¹⁷

adopted by the European Parliament opinion of 19 November 1998, Communication from the Commission dated 5 February 1999 to the Council, the European Parliament and the Economic and Social Committee, COM(1999)42

Consultations of the Commission on Software Patents 2000¹⁸, so called Independent study¹⁹ on the economic impact of the patentability of computer programs

¹⁷http://europa.eu.int/comm/internal_market/en/intprop/indprop/558.htm

¹⁸http://europa.eu.int/comm/internal_market/en/intprop/indprop/99.htm

¹⁹http://europa.eu.int/comm/internal_market/en/indprop/comp/studyintro.htm