



British &amp; Irish Law, Education and Technology Association

**20th BILETA Conference:  
Over-Commoditised; Over-Centralised; Over-  
Observed: the New Digital Legal World?**

*April, 2005, Queen's University of Belfast*

## Software emulation in the light of EU legislation

**Boštjan Berčič,**

University of Ljubljana, Faculty of Administration

### 1. Introduction

Emulators are programs that translate assembly language instructions between platforms, thus permitting software of one platform to be run on another. They are used primarily for emulation of no longer available hardware platforms (e.g. times past home computers such as Spectrum, Commodore etc.), but also for emulation of existing hardware (e.g. iMac emulation on PC). Software emulation has been for a long time limited to videogames emulation, but in the future it will increasingly find its use in other areas. One of them is digital archiving where original data and documents, which are archived on a long-term basis, need to be reproduced when the original hardware, software or data format are no longer supported by their vendors or developers. This reproduction might involve access to original data and documents and use of original software for these operations. For archived data, this may literally mean decades old software.

Many things have been asserted about legality of emulators (e.g. that they encourage piracy of copyrighted software, that they themselves constitute software copyright infringement, that emulating hardware is illegal, that emulation causes trademark dilution etc.). While some of these were decided by courts to be false (e.g. that emulating hardware itself is illegal), some claims remain credible (some emulators for example include operating system of the targeted platform, which may very well constitute copyright infringement).

There are a couple of important court rulings regarding this questions. For example, in *Sony v. Connectix*, Sony filed a lawsuit against Connectix, producer of VGS (Virtual Game Station), alleging that Connectix was infringing Sony's intellectual property rights. Court decided in 2000 that software emulation of hardware is legal and does not in itself represent infringement of copyright (lest software emulation contain verbatim copy of copyrighted software, e.g. emulated platform's OS). Even more, this ruling decided that reverse-engineering of targeted platform's operating system and other software (ROMs) for the purposes of building an emulator was considered fair use and thus allowed. Ruling established that there is no copyright in how computers are built – emulators are therefore legal. Another important case regarding legality of reverse engineering is *Sega v. Accolade*, which preceded and opened way for later *Sony vs. Connectix* ruling.

Most of these cases appeared outside EU and were ruled according to non-EU (mostly American) legislation. It is therefore interesting to see how these cases (and emulators-related issues on general) would be dealt within the framework of European legislation, especially in the light of EU Software Directive ([4]). This directive laid foundations for a unified approach to copyright-related issues of software across EU. Provisions of this directive should thus shed light on the debate about legality or illegality of software emulation.

Copyright, however, is not the only intellectual property issue that may arise in connection with emulators - there may be patent issues as well. Many parts of computer hardware can be patented (and are patented, for example many videoconsoles contain patentable parts), so software emulators

could very well infringe on intellectual property of patent holders. A new danger regarding emulators' potential infringement of patents is lurking on the horizon: software patents as proposed by a EU's Proposal for a directive on the patentability of computer-implemented inventions ([7], [8]) also referred to as Software Patents Directive) and as currently approved by European Patent Office. Besides patenting computer hardware, manufacturer of a computer system could, for instance, patent software as well, thus making emulation of it illegal.

This paper will try to consider some of emulators-related legal issues, primarily in the framework of extant and forthcoming EU legislation.

## 2. Emulators

Emulators are a piece of software that emulate one hardware platform on another. They do this by emulating in software the behaviour of the targeted platform chipsets (CPU, for instance). Ultimately, assembly language of the targeted platform is translated into the assembly language of the hardware that the emulator runs on. Functioning of the input and output devices are also emulated. Sometimes, parts of targeted platform's operating system are also employed (either emulated or included as verbatim copy). A question than naturally arises whether software emulation of a piece of hardware (e.g. a microchip) or another piece of software is legal.

## 3. Software emulation of hardware

Adversaries of emulators sometimes propose that (software) emulation of hardware is itself illegal – just as copying the copyrighted software would be. This argument was for example embraced by IDSA (now known as ESA) - that emulators are illegal because the hardware they emulate is proprietary and any emulation of that hardware without the consent of the manufacturer would constitute an illegal copyright infringement.

Legal arguments that were employed as to why emulation of hardware is illegal were: copyright violation, confidentiality violation and patent violation. Some of these are false, some might be true.

A software that emulates hardware functions, for example, does not violate copyright, since its structure is different from the mimicked platform and copyright is only concerned with the protection of expression of ideas, not ideas themselves. This for example came out of *Sony vs. Connectix* ruling (see [10]) where the judge held that Sony could prevail only if it had patented its hardware. Hardware whose functioning is a public knowledge (e.g. how some particular chip works) can be thus be emulated without any copyright violation.

Functioning of some hardware on the other hand is not in the public domain. If writers of emulator got that knowledge by breaching some confidentiality agreement, such emulation may be illegal. On the other hand still, even if some hardware design was not public knowledge, one could in principle still obtain such knowledge through reverse-engineering and write an emulator. The question than would be: is reverse-engineering of hardware for the purposes of obtaining knowledge about its functioning illegal? In *Sony vs. Connectix* Sony alleged that reverse-engineering of some of its custom-made chips was illegal, because it allegedly infringed against trade secrets. US Court of Appeals rejected this, thus effectively legitimizing reverse-engineering under fair-use doctrine.

Patent protection of hardware thus seems the only way that would make the allegation, that emulators are illegal, true. If hardware design or some pieces of it were patented (e.g. particular chips, I/O devices), than a realization of it in software could very well infringe on rights of the patent holders, since patents protect ideas, not expressions thereof. Patent protection of hardware, on the other hands, evokes other dilemmas. For example, is it possible to patent one particular chip, since chips nowadays are commonplace and the knowledge how to build them in public domain. It was historically possible to patent (first) microchip, but it might not be feasible to patent one today. Once

the patent protection for idea (e.g. microchip, printed circuit board) is over (20 years maximum), various expressions of that idea (e.g. particular microchips) might be protected by copyright (e.g. printed circuit boards, see [1] , pp. 143) or by sui generis rights (such as is indeed the case with semiconductors, which are protected under the terms of EU directive on semiconductor products with special semiconductor right, see [3] ) or they might not be protected at all (such as is the case with some I/O devices, e.g. computer mouse).

#### 4. Software emulation of software

There is no clear distinctive line between hardware and software. What we normally take to be hardware (a microchip) could just as well be realized in software (e.g. microprogramming, firmware). So the next question is whether a piece of software can legally emulate another piece of software. This is of great practical relevance for emulators usually don't emulate just hardware but software of the emulated platform as well (usually its operating system, BIOS). Other software that runs on targeted platform (e.g. applications, games) is normally not reverse-engineered and emulated but is used directly (in its direct version, such as when it is available on CD-ROMs or in its translated version, such as when ROMs are translated to ROM images).

Software is copyright protected. Literal copying is prohibited, so if one reverse-engineered the source code of the operating-system of one platform and port it to another, this would represent infringement of copyright (in fact, it would, 't even be an emulator, it would be a port). Reverse-engineering of a program and writing of another program with similar functionality is usually not considered copyright infringement, but it could be if we accept the interpretation that non-literal copying represents copyright infringement.

Similar questions arose in connection to (software) emulation of software. Does creation of a program with similar functionality to another program constitute copyright violation or not ? What emulation does is it produces same outputs (e.g. on screen) on a targeted platform when given the same inputs (e.g. ROMs). Emulation is therefore not literal copying (how the operating system is built) but non-literal copying (what the operating system does). Emulation is therefore not direct copyright infringement (because it is not a literal copy), but it could be an infringement if copying ideas of how something works (what it produces when given the same inputs) – non-literal copying would be deemed to be copyright infringement.

This is an important question of computer law on general, quite irrespective of what it entails for software emulators. Usually it is considered that copyright extends to non-literal copying as well, but there are some caveats with this general position.

First, this argument sometimes runs counter the normal intuition that copyright protects different expressions, not their underlying idea. Since one can achieve same outputs given same inputs with wildly different architectures, and if non-literal copying is considered copying as well, then this kind of protection goes in the direction of patent protection where the idea itself is protected, irrespective of its realization or expression.

Second, recent legal decisions seem to favour interpretation that non-literal copying is not as broad a category as it looked from the outset. In *Whelan Associates v. Jaslow Dental Laboratory* (see [12] ) it was held, for example, that not only literal copy of a code in some programming language is protected by copyright (as it traditionally was), but that ideas underlying this code could also constitute copyrightable matter. It was held, that if there are several ways of achieving some purpose, none of which is necessary to the purpose, then the way chosen is an expression (and as such copyrightable) and not idea. This decision considerably broadened the scope of the copyright protection – to the point that it started resembling patent protection. Not only code, but the underlying principles were held to be copyrightable as well, as long as there were many possible competing principles to achieve one specific goal. Only if the principles were so general that they didn't allow for alternative implementation, were they deemed an idea and not an expression. From

traditional point of view of what copyright should protect, this decision in fact represented a big detour. Not only (physical) expression (such as computer code) was copyrightable, but ideas themselves (e.g. the structure of the program, the sequence of operations) were proclaimed potential expressions and therefore protectable

Prof. Hollaar (see [6] ) writes with respect to this decision “By limiting the idea of a computer program to its highest-level purpose and considering everything else expression, the Third Circuit extended copyright protection to virtually every aspect of a computer program beyond its intended use. Data structures, program structures, ways of doing calculations, screen displays, output formats, and programming styles all could be protected. This is far broader protection than can be afforded new computer techniques under the patent laws.

Later decision in the USA (*Computer Associates International v. Altai.*, see [2] ) overruled *Whelan* decision and with it the interpretation according to which almost anything is copyrightable (cf. this with today’s situation regarding computer patents). Still, it didn’t rule out the possibility of non-literal copyright infringement. Instead, it proposed a new test for non-literal copying (AFC- Abstraction, Filtration, Comparison), according to which program code is first abstracted to several levels, not unlike reverse-engineering, non copyrightable elements are then filtered out and finally only the remaining copyrightable matter (if any) is compared to the original program, in relation to which disassembled program allegedly violates copyright. If their copyrightable matter is substantially similar, then there is copyright infringement.

Thus it is generally not considered to be a copyright violation if one software mimicks functionality of the other (e.g one spreadsheet program mimicking another spreadsheet program). It is also generally not considered to be a copyright violation if the user interface of one software has the same »look and feel« as another program (consider Windows skin, which can be used in Linux operating system). Similar considerations apply to the legality of emulation of a particular software. What is copied is a concept: what one program does, its functionality, how it behaves when some inputs are present. In a way, if emulated program was taken as a black box, and the emulator built from scratch, this would surely not be a copyright infringement since only functionality would be copied, and functionality can only be protected by patents (cf. Article 5(3) of the Software Directive which directly allows black-box reverse-engineering). The way most emulators are built in real life (through reverse-engineering), copyright issues do arise: for example, one usually has to copy, translate or decompile object code of a program back into its source code and all of these activities are copyright protected and withheld to the authors.

On the other hand, some emulators don't emulate targeted platform's operating system – sometimes they include a verbatim copy of it. This, however, does represent a clear violation of copyright.

## 5. Emulator creation through reverse-engineering

Even if the software emulators are (for the most part) legal, questions still arise as to whether the process of their development is legal or not. To write an emulator, one has to reverse-engineer the targeted platform. This means reverse-engineering of hardware (chips) as well as reverse-engineering of potential software (such as BIOS).

We have already established that reverse-engineering of hardware, unless that hardware is patented, is not disputable. How some machine works is not copyrightable matter, but it may be a patent matter. We therefore have to establish the legality of reverse-engineering of software (e.g. operating system or BIOS) for the purposes of building an emulator.

Software Directive lays forward principles for reverse-engineering viz. decompilation. According to the 1st paragraph of Article 6, this is allowed where »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:

- these acts are performed by the licensee or by another person having a right to use a copy of program
- the information necessary to achieve interoperability has not previously been readily available to these persons and
- these acts are confined to the parts of the original program which are necessary to achieve interoperability.

According to second paragraph of Article 6, decompilation is not allowed where information obtained through its application

- will be used for goals other than to achieve the interoperability of the independently created computer program;
- will be given to others, except when necessary for the interoperability of the independently created computer program
- will be used for the development, production or marketing of a computer program substantially similar in its expression, or for any other act which infringes copyrights.«

The question here is whether any of these provisions present a problem for the emulation of software.

First, any decompilation is allowed only to achieve the interoperability of an independently created program. In other words (and in accordance with directive's preamble), in order to write an interface, which is a special computer program, designed to achieve interconnection and interaction between different (software and hardware) components of a computer system. Reverse-engineering of software in order to build an emulator is then allowed according to the Software Directive if emulators fall in the category of interfaces. This is not completely straightforward, but one can argue, that emulators are indeed interfaces connecting various (software and hardware) parts to work together. Emulator allows one to run software written for other platforms on the platform it was written for, thereby bypassing targeted platform's operating system. Emulator thus works as an interface between native hardware platform (platform it was written for) and software written for other platforms. It indeed provides interoperability and can be thus be argued to be an 'interface'.

Then there are a couple of conditions which are, some more, some less trivially attainable. First, developer who wants to reverse-engineer some piece of software in order to write an emulator and not transgress the law (in EU, from 1993 onwards at least) has to be a legal owner of a copy of a program. Second, the information necessary to achieve interoperability must not be otherwise available. This means that it is not allowed to reverse-engineer something if the information about its design and functioning is separately available – this provision is due to the worry that a person performing reverse-engineering might get a hold of more information than absolutely necessary and would therefore potentially transgress copyright. This might present a bit more of a problem because documentation is sometimes available, but emulators are, for technical reasons, usually written through reverse-engineering even though documentation might be available. With respect to this condition, emulator developers will thus have to prove that the documentation was not available or that it was not sufficient to produce the emulator. Third, one is legally allowed to reverse-engineer only those parts of computer program which are necessary to achieve interoperability. Fourth, decompilation must not be used for purposes other than to achieve the interoperability. Fifth, information obtained through decompilation must not be given to others, except for the interoperability of the independently created computer program. Proprietary emulators (such as UltraHLE, VGS) will not have problems with these, but so will not those which are open source because there the information obtained is given away to third parties, but presumably (under appropriate open source licences) only for further development of the emulator in question. Sixth, this information must not be used to develop a computer program substantially similar in expression to the one being reverse-engineered. Emulator is a software which is similar in idea to the emulated software, but not necessary in expression. If reverse-engineering is used to write a substantially

similar program in expression (for example reverse engineering of BIOS to write slightly modified BIOS), than it might very well represent copyright infringement. Here, question of what is expression and what idea in the computer program must be answered again. In its standard sense, the code of the emulator must not resemble too much the code from the program being reverse-engineered – this can be for example achieved with 'clean room' procedure, where one team of experts decompile the program and write the specifications for the emulator and the other team of experts, who never saw original code, then writes the emulator.

Emulation through reverse-engineering is therefore legal according to the Software Directive, but this legality hinges on some conditions. If these are met, emulators are indeed legal according to EU legislation.

## 6. Dumping ROMs

Similar problem arises in connection with ROMs, pieces of hardware, which store computer programs (usually games). In order to build emulator, although not absolutely necessary, developers usually use these ROM images to see, whether developed code behaves so as to mimic the original platform. This would present no problem if CD-ROMs would be used (by its legal owner) – instead of using them on the original platform, emulator would be used. Hardware ROMs (cartridges, arcade cabinet carts) are more complicated in this respect, because they have to be translated into software. ROM image is a file, obtained through ROM dumping, which contains exactly the program, stored on ROM. This dumping involves the process of copying and every copying requires the right to do so. Here, recourse to Article 5(2) of the directive can be made. It says that the »making of a back-up copy by a person having a right to use the computer program may not be prevented by contract insofar as it is necessary for that use.« Question here is, whether dumping ROMs is necessary for their use. It might be if ROMs or the machine they are supposed to run on physically wear out – than it might be ok to make a (software) copy of them.

First thing to be answered here is what legal category covers ROM dumping. Is it simple copying (copying of machine instructions from ROM's memory to the binary file) or is it a translation.

For if it is simple copying, than it falls under Article 5(2) of the directive which specifies when back-up copies can be made. Copying is under specified circumstances allowed to legal owners. Article 5 (2) of the directive says that »the making of a back-up copy by a person having a right to use the computer program may not be prevented by contract insofar as it is necessary for that use«. Similar provisions exist in US law (fair use doctrine)

If dumping ROMs is considered translation, on the other hand (for example from hardware to software), than we have to make recourse to the articles 4 and 5 of the directive, which say that (Article 4) »the rightholder should have the right to do or to authorize the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof, without prejudice to the rights of the person who alters the program« and (Article 5) these acts »shall not require authorization by the right holder where they are necessary for the use of computer program by the lawful acquirer in accordance with its intended purpose, including for error correction.« Is it then of crucial importance to establish whether such translation (from ROMs to binary files) is in accordance with the intended purpose of the program. This issue is not clear, at least not in the general sense and it might have to be decided on a case by case basis.

We can distinguish two examples here. One is dumping ROMs to binary files with sole purpose to execute them on a simulator (e.g. for playing games) and the other is dumping them with ulterior motive - in order to test the functioning of the emulator. In both cases, this dumping can be put into different legal categories. With users who dump (their own) ROMs in order to play them on an emulator, this dumping is either copying, according to article 5(2), or translation, according to article 5(1) which has to be in accordance with the intended purpose of the program. With emulator developers (who also have to own these ROMs), this dumping is either copying in accordance with



article 5(2), or translation according to article 6 (decompilation) which is allowed under specified circumstances. These circumstances are effectively the same we saw for dumping BIOS, except that they are more easily satisfied in the case of dumping ROMs, because there are no problems now with potential availability of the information necessary to achieve interoperability according to article 6 (1) (b) – videogames are usually not supplied with this kind of information and no problems usually arise with usage of such information to develop, produce or market a computer program substantially similar in its expression in accordance with article 6 (2) (c) – emulator developers are usually not game developers.

## **7. Other allegations against emulators**

### **Inciting piracy**

Emulators were also said to incite piracy, especially in connection with two issues. The first is translation of ROMs to ROM images and potential divulgation of such files. While physical ROMs can't be easily multiplied, their ROM images can be quickly distributed with almost zero cost of copying. Emulator developers were thus deemed to incite piracy and therefore be guilty of contributory infringement.

This argument, however, is applicable in American legal theory only. EU directive does not include provisions for contributory infringement, it talks only about direct infringement. According to article 7 (1) (a) »any act of putting into circulation a copy of a computer program knowing, or having reason to believe, that it is an infringing cop« shall be punishable. Similarly, according to article 7(1) (b) »the possession, for commercial purposes, of a copy of a computer program knowing, or having reason to believe, that it is an infringing copy« shall be punishable.

### **Circumvention of antipiracy protection**

The second issue is circumvention of the antipiracy protection that is from time to time embedded in the hardware (such as dongles and other counter-piracy devices). Emulators usually don't emulate this protection (as their principal raison d'être is exactly to get rid of hardware) so they can be said to make piracy easier.

According to article 7 (1) (c) (Special measures of protection), national legislations shall provide appropriate remedies against persons committing »any act of putting into circulation, or the possession for commercial purposes of, any means the sole intended purpose of which is to facilitate the unauthorized removal or circumvention of any technical device which may have been applied to protect a computer program«. This, however, shall be construed without prejudice to the provisions of Articles 4, 5 and 6. EU directive therefore does allow one to remove technical devices which protect computer programs if this removal is for the purposes of decompiling, reverse-engineering and creating back-ups.

### **Brand tarnishment**

Emulators were also accused of brand tarnishment. In connection with this, some emulators offer originally unthought-of user-experience enhancement (or hindering for that matter). Some emulators, for example, allow one to save position in the game, which is an option unavailable in most original games. Or, for example, user may be allowed to fast-forward the game to some pre-specified section – an option also unavailable in the original game. Elsewhere, emulators might not correctly reproduce user experience (e.g. missing sound, jerky gameplay etc.). All these might lead to trademark dilution.

### **Business model destruction**

There were also allegations that emulators are destroying business model of the producers of the original software (mostly video games manufacturers). First, it is usually much cheaper to buy (or

even acquire for free) an emulator than to buy the original platform that it emulates. Second, emulators provide backward compatibility, which is not a standard in the gaming business. Once the video console comes off age, it is supplanted by another one which is not backward compatible. Customers wishing to play old games are forced to buy them anew for the new platform because the older one is no longer supported – this business model is also thwarted by the emulators. These however are more business than legal issues and I will not further explore them in this paper.

## 8. Conclusions

I tried to illustrate legal issues regarding software emulation and how they would be resolved according to existing EU legislation in the field, above all the EU Software Directive and (to lesser extent) the proposed Directive on software patents. Relevant cases and court decisions regarding emulators almost all emanate from the American jurisprudence. These rulings, however, would not be all that different according to the EU legislation, because there are counterparts to American precedents in the EU directives. What was established to be fair-use in the USA, for example, is protected by the decompilation right in the EU. On general, one can conclude that a comparable legal framework is established in the EU and USA regarding software-related legal issues and that, although most cases involving emulators were decided in the USA, European courts would probably reach decisions much along the same lines as their American counterparts.

---

## References

1. Bainbridge D.: Introduction to Computer Law Fourth Edition, Pearson Education Limited, Harlow, 2000
2. *Computer Associates International v. Altai*. {FN53: [982 F.2d 693](#), [23 USPQ2d 1241](#) (2d Cir. 1992)}
3. Council Directive 87/54/EEC of 16 December 1986 on the legal protection of topographies of semiconductor products (Official Journal L 024 , 27/01/1987 P. 0036 – 0040)
4. Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs, OJ L 122, 17/05/1991 p.0042-0046
5. Ferrera, Lichtenstein, Reder, Bird, Schiano, CyberLaw Text and Cases (Thomson South-Western West, Mason, 2004)
6. Lee A. Hollaar: Legal Protection of Digital Information, BNA Books (October 1, 2002)
7. Proposal for a Directive of the European Parliament and of the Council on the Patentability of computer-implemented inventions, General Secretariat of the Council, Brussels, 24 May 2004
8. Proposal for a Directive of the European Parliament and of the Council on the Patentability of computer-implemented inventions, Brussels, 20.02.2002
9. Reed and Angel, Computer Law (5<sup>th</sup> ed, Oxford: OUP, 2004)
10. Sony Corporation of America v Connectix Corp., 48 F. Supp. 2d 1212, 1216 (N.D. Cal. 1999)
11. Tsiricos Takis: One Flew Over Pac-Man's Nest An Authoritative Study on Emulators and the



Problems that they Pose in Relation to Intellectual Property Rights, University of Newcastle Upon Tyne, 1999

12. *Whelan Associates v. Jaslow Dental Laboratory*. {FN42: *Whelan Associates v. Jaslow Dental Laboratory*, [797 F.2d 1222](#), [230 USPQ 481](#) (3d Cir. 1986)}