

Copyright, Interfaces, and a Possible Atlantic Divide

by **Simonetta Vezzoso**, Trento*
PhD, University of Trento, Italy

Abstract: Recent copyright cases on both sides of the Atlantic focused on important interoperability issues. While the decision by the Court of Justice of the European Union in *SAS Institute, Inc. v. World Programming Ltd.* assessed data formats under the EU Software Directive, the ruling by the Northern District of California Court in *Oracle America, Inc. v. Google Inc.* dealt with application program-

ming interfaces. The European decision is rightly celebrated as a further important step in the promotion of interoperability in the EU. This article argues that, despite appreciable signs of convergence across the Atlantic, the assessment of application programming interfaces under EU law could still turn out to be quite different, and arguably much less pro-interoperability, than under U.S. law.

Keywords: Software Directive, Interoperability, Interfaces

© 2012 Simonetta Vezzoso

Everybody may disseminate this article by electronic means and make it available for download under the terms and conditions of the Digital Peer Publishing Licence (DPPL). A copy of the license text may be obtained at <http://nbn-resolving.de/urn:nbn:de:0009-dppl-v3-en8>.

This article may also be used under the Creative Commons BY-NC-ND 3.0 Unported License, available at <http://creativecommons.org/licenses/by-nc-nd/3.0/>

Recommended citation: Simonetta Vezzoso, Copyright, Interfaces, and a Possible Atlantic Divide, 3 (2012) JIPITEC 153, para. 1.

A. Introduction

- 1 In the field of intellectual property law, interoperability refers mainly to the ability of information technology products to communicate, i.e. to exchange signals and data. From a user's perspective, products or systems are considered to be interoperable if they can work together. Beyond IT markets, assuring the compatibility of products with those of other brands can also be essential in order to assure the satisfactory working of competition processes.
- 2 Interoperability has obvious intrinsic value and, therefore, it is in the interest of society to support it. Seemingly, there are often private incentives at work that induce undertakings to spontaneously release the relevant interface information. For instance, it can be in the interest of a platform owner to share the rules of interconnection between her core technological building blocks and the surrounding ecology in order to promote the development of sets of complementary products and services and,

by doing so, increase the overall attractiveness of the platform. From the perspective of a business, decisions about the desired level of interoperability are both technology and market based, ultimately depending on the firm's expectations about its ability to create and capture economic value. In this respect, practices relating to interoperability can have positive or negative effects on the level of healthy competition, both *infra-* and *interplatform*.¹ Interoperability within a single platform is also called *vertical* interoperability, as opposed to the *horizontal* interoperability across platforms. Of course, horizontal interoperability is expected to have a decisive impact on the level of interplatform competition, especially when users of one platform stick to it because they cannot move their data and applications to another, "better" platform. The issue of vertical interoperability, and the denial of it, relates to the relationship between the platform owner and third party developers.²

- 3 The most significant *ex-ante* intellectual property rules on interoperability in IT markets are presently located in the area of copyright law.³ Ideally, copyright law should provide market participants with clear indications as to the ability to access and use interoperability information.
- 4 Unsurprisingly, perhaps, this approach delivers only partially satisfactory results. With regards in particular to Europe, the Software Directive has some indefinite language on the crucial issue of interoperability, and courts have been rather slow in delivering the necessary interpretative rulings. More than twenty years after the enactment of the EU Software Directive, the Court of Justice of the European Union (“CJEU”) in *SAS Institute, Inc. v. World Programming Ltd.* (C-406/10) (hereinafter *SAS v. WPL*), articulated for the first time an explicit ban on copyright protection in what are some of the building blocks of software, such as programming language and data formats. Moreover, the decision denies copyright protection to the functional effects of software. On the whole, the recent CJEU’s pronouncement provides those conducting the emulation of existing programs with much less shaky legal ground to stand on regarding their activities (Section 2, *infra*). The accrued legal certainty could have positive effects on competition in software markets. On the negative side, still unanswered under EU law are very important issues regarding interoperability, such as those decided in the U.S. *Oracle America, Inc. v. Google Inc.* (C 10-3561) (hereinafter *Oracle v. Google*) (Sections 3 and 4, *infra*).
- 5 Computer programs are protected as literary works under copyright law, within the meaning of the international Berne Convention for the Protection of Literary and Artistic works. Therefore, the written program code (both source and object code) is protected under copyright law by analogy with other literary works such as the text of a novel or poem. Based on the legal doctrine known as the idea/expression dichotomy, it is well understood that the scope of copyright protection for computer programs is limited to the expression of the program. The main rationale behind the idea/expression dichotomy is that it is socially desirable to allow for the free use of the fundamental building blocks (the “ideas”) of knowledge production in the area of copyright protection.
- 6 The idea/expression distinction manifested itself as a solid theme in support of the proposal of copyright protection for computer programs in the EU put forth in 1989.⁴ Article 1(2) of the subsequently enacted EU Software Directive⁵ states that “[i]deas and principles which underlie any element of a computer program, including those which underlie its interfaces, are not protected by copyright.” However, what exactly constitutes the “idea” of a literary work, excluded as such from copyright protection, continues to stir very lively debates in Europe and elsewhere.
- 7 The question can be usefully framed in terms of the extent to which copyright infringement can consist in non-literal copying of the work. The concept of non-literal copying derives itself from traditional copyright law, and indicates that infringement in a work can take place even though little or nothing of the actual work was used. Thus, for instance, the plot of a novel may be protected by copyright and infringed even when there is no evidence that literal copying (i.e. actual copying of the text) has occurred. With respect to the non-literal copying of a computer program, however, an extra layer of complexity is added because of the dual nature of these works, that is “textual works created specifically to bring about some set of behaviors.”⁶ Therefore, whereas a computer program *behaves*, more traditional works protected by copyright, such as books, manifestly do not, and a too broad interpretation of the “plot analogy” could inappropriately extend copyright protection to the functional aspects (the behaviour) of a computer program.

B. SAS v. WPL and the EU Software Directive

I. Non-literal copying of computer programs

- 8 The recent *SAS v. WPL* ruling by the CJEU touches upon a whole set of crucial legal issues concerning the object and scope of the copyright protection of computer programs in the EU, thus extending its relevance well beyond interoperability issues. *World Programming Ltd.* (“WPL”) had carefully studied very successful analytical software developed by the SAS Institute (“SAS”), which enabled users to write and run application programs to perform data processing and analysis tasks. Based on that, WPL created an alternative computer program (system/platform) allowing users to execute application programs already written in the SAS language, and ensuring that the same inputs would produce the same outputs. WPL’s explicit goal was to enable SAS customers to run their own application programs written in the SAS language on WPL’s alternative platform.
- 9 The preliminary reference from the English High Court⁷ gave the CJEU the overdue opportunity to start clarifying, among other things, the scope of protection against non-literal copying of computer programs. Importantly, in the *SAS* decision it could not be established that the defendant in the national copyright infringement case, WPL, had access to object or source code and/or carried out any de-

compilation of the object code. WPL merely studied how the SAS program worked by performing “black box”⁸ analysis and reading the SAS manuals. WPL then wrote its own program to emulate the SAS program’s functionality, i.e. the specific processing and analysis tasks performed by the program. The subject-matter “taken” by the defendant WPL, which was at the basis of the allegation of copyright infringement, included the functionality of the SAS computer program, its language, and data formats.

- 10 In answering the questions posed by the national judge, the CJEU basically follows the opinion⁹ previously delivered by its Advocate General (“AG”) Yves Bot. The Court, however, seems to depart from AG Bot’s opinion on a very sensitive issue, as we will see in the following.
- 11 AG Bot first iterated the important role played by the internationally acknowledged idea/expression dichotomy in the area of copyright protection of computer programs. According to AG Bot, the reason underlying this core copyright principle was that “the originality of a work, which gives access to legal protection, lies not in an idea, which may be freely used, but in its expression” (para. 44). It is perhaps necessary to remind at this point that the Software Directive deliberately left to the courts the task of ascertaining the boundaries of this central dichotomy under EU law, but this, unfortunately, has created broad areas of uncertainty and possibly missed more ambitious harmonization objectives.
- 12 With regard to the scope of copyright protection for computer programs, AG Bot suggested that this is “conceivable only from the point at which the selection and compilation of those elements are indicative of the creativity and skill of the author and thus set his work apart from that of other authors” (para 48). Like a book author, the programmer “selects the steps to be taken and the way in which those steps are expressed gives the program its particular characteristics of speed, efficiency and even style” (para. 47).
- 13 As to the object of protection according to the Software Directive, AG Bot referred to Case C-393/09 *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v Ministerstvo kultury*,¹⁰ decided in December 2010, in which the CJEU stated that protected is “the expression in any form of a computer program which permits reproduction of that program in different computer languages, such as the source code and the object code...from the moment when its reproduction would engender the reproduction of the computer program itself, thus enabling the computer to perform its function” (para. 49). The AG Bot therefore came to the conclusion that “the protection of a computer program is not confined to the literal elements of that program, that is to say, the source code and the object code, *but extends to*

any other element expressing the creativity of its author” (para. 50, emphasis added).¹¹

- 14 The English reference, however, required the Court to take position on issues going well beyond its 2010 ruling, i.e. concerning the copyright assessment of functionality of a computer program, its language, and data formats.

II. Functionalities

- 15 With respect to the functionalities of a computer program, AG Bot defined the latter as “the set of possibilities offered by a computer system, the actions specific to that program” (para. 52). He concluded that the functionalities “as such” cannot form the object of copyright protection under Article 1(1) of the Software Directive (para. 53). According to AG Bot, the multitude of actions “dictated by a specific and limited purpose” (e.g. performing online hotel bookings), are similar to an idea, and therefore other computer programs should be able to offer the same functionalities (para. 54). Eligible for copyright protection are instead “the many means of achieving the expression of those functionalities,” i.e. “the way in which the elements” composing a computer program “are arranged, like the style in which the computer program is written” (para. 55).
- 16 The CJEU squarely endorses the Advocate General’s statement according to which, to accept that the functionality of a computer program can be protected by copyright, “would amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development” (para. 40; AG Bot para. 57). It should also be noticed that the language employed by the CJEU and AG Bot strongly resonates with previous US decisions stating that copyright protection of computer programs should not extend to the program’s functionality because this would hamper scientific progress, ongoing innovation and competition in the marketplace.¹² This in turn mirrors the concern expressed by the already recalled TRIPS Agreement and WCT provisions that copyright protection for computer programs should not extend to “ideas, procedures, methods of operation, and mathematical concepts.”¹³ Moreover, the CJEU refers to the explanatory memorandum to the Proposal for a Software Directive,¹⁴ pointing out that by protecting “only the individual expression of the work,” other authors would have had ample latitude “to create similar or even identical programs provided that they refrain from copying” (para. 41). Interestingly, the CJEU does not seem to conform to the exact, and arguably more restrictive, language employed by AG Bot, who, as we have seen above, referred to “functionalities as such.”

III. Programming language and data formats

- 17 Beyond SAS computer functionalities, the subject-matter “taken” by the defendant WPL included SAS programming language and data formats. Therefore, the CJEU had to further decide whether the programming language and the format of data files were protectable by copyright law, i.e. whether they constituted a form of expression of the program.
- 18 With respect to “programming language and the format of data files used in a computer program in order to exploit certain of its functions,” the Court concludes that they do not “constitute a form of expression of the program and therefore are not protected by copyright in computer programs according to the EU Software Directive” (para. 46). Referring once more to *Bezpečnostní softwarová asociace*, the Court does not exclude the possibility that “the SAS language and the format of SAS Institute’s data files might be protected, as works, by copyright” under general copyright rules if “they are their author’s own intellectual creation” (para. 45).
- 19 In his Opinion to the CJEU, AG Bot argued with regard specifically to SAS programming language, that this is “a functional element which allows instructions to be given to the computer...made up of words and phrases known to everyone and lacking originality.” Thus, concluded the AG, the “programming language must be regarded as comparable to the language by the author of a novel. It is therefore the means which permits expression to be given, not the expression itself” (para. 71).
- 20 The assessment of interfaces in the form of data formats put forth by the CJEU in SAS offers further and most needed clarification, based as it is on the legal interpretation of less than crystal clear language of the EU Software Directive.
- 21 To understand this, it should first be considered that interface specifications are concrete, in written form, and normally form part of the code. Second, despite often challenging constraints, there can still be some room for programmer’s choices among functionally identical ways of formulating the interface specifications, in that case satisfying the Software Directive’s originality requirement. Third, the Directive does not state that copyright protection is unavailable to interfaces as such. Nevertheless, various recitals and articles would more or less explicitly promote interoperability.¹⁵
- 22 With respect specifically to data formats, the referring UK High Court asked whether it is an infringement for a program to be implemented in such a way that enables that same program to read and write data files that are in the format devised by another program. It is perhaps useful to recall at this point that WPL had reportedly obtained the interface information merely by studying how the SAS system operated. Specifically, WPL worked out enough of the data formats employed by the SAS system to be able to write a new source code that read and wrote data files written in that format, and by that, achieving interoperability with SAS users’ computer programs (scripts).
- 23 As anticipated above, the CJEU denied copyright protection to data formats under the Software Directive. The Court in particular reminded that “WPL did not have access to the source code of SAS Institute’s program and did not carry out any decompilation of the object code of that program” (para. 44), thus clearly differentiating its own assessment of the relevant facts from AG Bot’s imprecise view that WPL had performed an act of decompilation (AG Bot, para. 83). Doing so, the Court also “neutralized” the very restrictive interpretation of the decompilation provision put forth by AG Bot, who pointed *inter alia* to the requirement for the licensee to demonstrate the “absolute necessity” of its actions.¹⁶
- 24 However, both with regard to the SAS language and the format of SAS Institute’s data files, the CJEU concluded that they might be protected as works by copyright under general copyright rules if they are their respective author’s own intellectual creation (para. 45). While this last remark would seem to make the Court’s holding on the issue of this type of interfaces under EU law less clear-cut than some would have wished, it is nevertheless hard to dispute that the Court’s explicit reference to the “general” copyright Directive¹⁷ should not be read as authorizing the sweeping vacation of the fundamental copyright principle, enshrined in international copyright law, that ideas cannot be protected.
- 25 While the SAS v. WPL decision should be welcomed,¹⁸ it still remains to be seen if the language chosen by the CJEU will make the SAS decision sufficiently operational, especially with respect to the idea/expression dichotomy as applied to computer programs, considering that this case is bound to be carefully deciphered by other courts and various stakeholders in the months and years to come.
- 26 Needless to say, of particular importance are the possible repercussions of the SAS v. WPL decision on interoperability issues that were not expressly covered by this ruling, such as the copyright protection of application programming interfaces (“APIs”). The remaining parts of this article briefly consider how a District Court on the other side of the Atlantic recently decided the question of the copyrightability of APIs. This article concludes that it is far from inevitable that a court on this side would come to a similar conclusion under EU law.

C. Oracle v. Google and the Copyright Protection of APIs

I. Java's fork and the copyright dispute

- 27 In 2010, shortly after having acquired Sun Microsystems, Inc., Oracle sued Google for infringement of Java-related copyrights and patents.¹⁹ The *Oracle v. Google* case turned out to be a complex dispute, finally decided by the San Francisco Judge William Alsup on June 20, 2012, only a few weeks after the CJEU issued the SAS ruling considered in the previous Section.²⁰
- 28 Most interesting for our purposes, Judge Alsup had to, in particular decide whether the *structure, sequence, and organization* (“SSO”) of APIs were protected under US copyright law. It should first be reminded that the phrase structure, sequence, and organization does not appear in the U.S. Copyright Act. Nevertheless, it has become a sort of metaphor for non-literal copying of computer programs, whereas the more general issue under copyright law, as mentioned before, originally emerged with respect to traditional works such as books.²¹
- 29 Thus, with regard specifically to the non-literal copying of software, the phrase “SSO” captures the thought that the structure or order could, under specific circumstances, lie on the expression side of the idea/expression dichotomy. Specifically, in the case at issue it was alleged that Google's Android platform had copied the SSO of the overall code for 37 APIs packages of Oracle's Java platform. Interestingly enough, it was the first time that a decision by a U.S. court waded into the issue of the copyrightability of APIs. Judge Alsup came to the conclusion that Google had not infringed Oracle's Java-related rights, in particular because Oracle could not claim any copyright to the SSO of Java APIs.
- 30 Similar to the SAS data formats, APIs are interfaces that allow software programs to communicate with each other. In particular, the Java APIs specify the behavior of program modules, while the so-called class library is the compiled object code implementing API specifications. By consulting the respective APIs, programmers can make use of specific program modules without having to know the details of the modules' inner workings. Communication between software programs is achieved by following the same set of specifications.
- 31 Java currently has 209 API “packages” for the Java Standard Edition, compared to the 8 packages it had in 1996 when Java was first released. API design is a very complex task, involving difficult choices and

requiring significant expertise and time.²² Java API “packages” are broken into “classes” and “methods.” Put differently, they are articulated in pre-written programs (classes) carrying out subroutines (methods). Whereas APIs change over time, the Executive Committee of the Java Community Process has approved only three changes of the Java language so far.

- 32 The Android platform for mobile devices was launched in 2007, most likely as a reaction to the threat that the emergence of smartphones posed to Google's core business model. The Android platform uses the Java programming language, the Dalvik virtual machine, and provides 168 of its own API packages, many of which have the same functions of Java APIs, but different design. However, Google replicated the SSO of 37 Java API packages, possibly those that “typical” Java programmers would expect to be callable by the same names used in Java and which were key to mobile devices. Specifically, Google used different code to implement the classes and methods of 37 Java API packages, but replicated their exact names and functions. In total, Google wrote – or acquired – 97 percent of the new code, whereas the remaining 3 percent consisted of replicated overall name organization and functionality.
- 33 As a result, applications written in the Java language could call the 37 sets of functionalities in Android by the same names as those used in Java. Computer programs already written for the Java platform could run on Android to the extent that they call functionalities of those 37 sets. It follows that computer programs written to run on Java were, to a certain extent, also able to run on the Android platform. Conversely, programs written for the Android platform were not fully compatible with the Java platform. In this respect, the compelling concept “write once, run anywhere,” did not hold true anymore for programs written in the Java language.

II. Disentangling the APIs

- 34 With regard to the exact nature of the Java SSO replicated by Google, to the extent that this turned out to be relevant to their copyright assessment, Judge Alsup drew a distinction between, first, the declaration or method header lines; second, the method and class names; and third, the grouping pattern of methods (p. 5 ff.).
- 35 As to the first, the San Francisco District Judge concluded that “(u)nder the rules of Java, they *must be identical* to declare a method specifying the *same* functionality – even when the implementation is different” (p. 3, emphasis in the original). Put differently, since “every method and class is specified to carry out precise desired functions,” it follows that the header (non-implementing code) line of code

“stating the specifications must be identical to carry out the given function” (p. 5). Therefore, since there is only one way to express an idea or function, this under U.S. merger doctrine must be free for everybody to use (p. 35). According to that well-established doctrine, “courts will not protect a copyrighted work from infringement if the idea underlying the copyrighted work can be expressed in only one way, lest there be a monopoly on the underlying idea.”²³

- 36 With regard to the second, i.e. method and class names, and third, i.e. the way in which the methods are grouped, there was not the same “functionality constrain” as with the header lines. Nevertheless, Judge Alsup decided that there was no copyright infringement. Specifically, with regard to method and class names, the legal argument presented by Judge Alsup was that “copyright protection never extends to names or short phrases as a matter of law” (p. 2).
- 37 The assessment of the way in which methods are grouped turned out to be perhaps the most delicate aspect of the whole decision. In fact, Judge Alsup basically agreed with Oracle’s assertion that in Android the methods could have been arranged in ways different from Java’s groupings and yet offer the same functionality. However, according to the U.S. District Judge, “while the overall scheme of file name organization resembles a taxonomy, it is also a command structure for a system or method of operation of the application programming interface” (p. 37, emphasis in the original), “a long hierarchy of over six thousand commands to carry out pre-assigned functions” (p. 38). The command structure is “a utilitarian and functional set of symbols, each to carry out a pre-assigned function.” It therefore qualifies as “system or method of operation” under Section 102(b) of the U.S. Copyright Act, and could not be copyrighted, even if it is original or even creative.
- 38 According to Judge Alsup, the interoperability argument would provide further support to the overall analysis of the Java grouping of methods under copyright law, and in particular, its character as a functional system or method of operation. The “fragmentation” among platforms lamented by Oracle during the trial, i.e. of the flawed interoperability between the Android and the Java platforms, was due to the fact that only Java-based applications using exclusively the replicated parts of the 37 API packages were Android-compatible. In case Java-based code needed a 38th package, it could not run on the Android platform. Therefore, insofar as the command structure for the 37 Java API packages was replicated in Android, third-party source code relying on those packages could run on the Android platform, by that achieving a certain level of interoperability/compatibility. In fact, those APIs whose organization differed from Java APIs would not have been interoperable, “for the name structure itself dictates the

precise form of command to call up any given method” (p. 12).

- 39 According to U.S. case law, interface procedures that needed to be replicated in order to achieve interoperability were “*functional* requirements for compatibility” (emphasis added), and as such were not copyrightable under Section 102(b). Finally, full interoperability was not relevant to the Section 102(b) analysis (p. 39).

D. Conclusion: Legal Assessment of APIs under the EU Software Directive?

- 40 Inevitably, even a cursory look across the Atlantic raises the question how a similar or identical case would have been decided by a national judge applying EU law. In the following it will be explained that, despite the influential CJEU decisions in *SAS v. WPL* and *Bezpečnostní softwarová asociace*, there is still no obvious answer to this important issue concerning interoperability under EU law. But it can be reasonably expected that, given the Software Directive’s clear intent to foster interoperability, courts will treat copyright infringement claims involving APIs with the necessary care.
- 41 Undoubtedly, the part of the *SAS v. WPL* decision concerning computer program’s functionalities illustrates an interesting convergence between the software copyright regimes across the North Atlantic, also consistent with international treaty obligations. In fact, both the Agreement on Trade-Related Aspects of Intellectual Property Rights and the World Copyright Treaty state that copyright protection for computer programs should not extend to ideas, procedures, methods of operation, or mathematical concepts. As reminded both by the CJEU in *SAS* and by Judge Alsup in *Oracle v. Google*, ideas and functions cannot be monopolized by way of copyright protection.
- 42 Of course, the idea/expression dichotomy and the banning of functionalities from copyright protection apply to methods in APIs as well. Thus, as explained in nicely plain language by the U.S. District Court judge, everybody is free to write her own code to carry out a function (i.e. comparing two numbers and returning the greater) “so long as the implementing code in the method body is different from the copyrighted implementation” (p. 34). Especially after the *SAS* decision, there should be little doubt that, under EU law, a court applying the crucial idea/expression distinction to APIs is bound to come to roughly the same conclusion as the U.S. judge. Moreover, AG Bot also drew a clear distinction between data files as “blank forms which are to be filled with the customer’s data by the SAS System and which

contain specific locations in which particular information must be written in order for the system to be read and write the file correctly” (para. 79) and the “the elements which create, write and read the format of said SAS data files” which are “expressed in source code in the program,” and concluded that SAS source code implementing the data format could be copyrightable (paras. 81-82). Similarly, one could argue that the idea represented by the API is not copyrightable, whereas the source code implementing the API in principle is protected.

- 43 However, as seen above, Google went further, and its actions raised in particular the question whether the SSO of the API could be copyrightable.
- 44 First, Google replicated the *method specification* as set forth in the declaration. However, Judge Aslup ascertained that under the Java rules, the declaration must be precise; otherwise it would carry out some other function. Therefore, to this “part” of the broader API issue, the judge decided to apply the merger doctrine, which bars the claiming of copyright ownership in an expression if there is only one way to formulate something.
- 45 Under EU law, it is not sure whether the conclusion on this specific point would be the same. For once, the status of the merger doctrine under EU law is considered to be uncertain.²⁴ It should be reminded, however, that the already mentioned Software Directive’s original proposal²⁵ contained a rather explicit reference to that doctrine where it said that “(i)f similarities in the code which implements the ideas, rules or principles occur as between inter-operative programs, due to the inevitability of certain forms of expression, where the constraints of the interface are such that in the circumstances no different implementation is possible, then no copyright infringement will normally occur, because in these circumstances it is generally said that idea and expression have merged.” Moreover, the CJEU in *Bezpečnostní softwarová asociace*, while discussing the issue of “additional” protection for graphical user interfaces under general copyright law, held that “where the expression of those components [which form part of the graphic user interface, SV] 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” (para. 49). The Court’s explanation, however, did not refer to the idea/expression dichotomy, but to the lack of originality, when in the following paragraph it explained that “(I)n such a situation, the components of a graphic user interface do not permit the author to express his creativity in an original manner and achieve a result which is an intellectual creation of that author” (para. 50).²⁶
- 46 Going back to the U.S. decision, Judge Aslup further held that the copying of the *method and class names* could not give rise to copyright liability, for under U.S. law, names and short phrases cannot be copyrighted in the first place. Again, under EU law the conclusion on this aspect of the broader API issue could turn out to be different than in the U.S. In *In-fopaq International A/S v. Danske Dagblades Forening* the CJEU held that copying short phrases could invite copyright liability so long as those fragments demonstrated the author’s intellectual creation.²⁷
- 47 At least as uncertain under EU law would be the situation with regard to the copyright assessment of grouping of methods as found in the Java APIs. It should first be reminded that the drafters of the Software Directive included preparatory design materials within the concept of computer programs. It follows that it would normally give rise to a copyright infringement for a program to be based on another program’s preparatory design materials. This inclusion can be read as a revealing indication that the drafters of the Software Directive intended to grant copyright protection to the detailed SSO of the internal design of program writings.²⁸ Further guidance could be drawn from the CJEU holding that, while elements described in the SAS manual—including keywords, syntax and commands—could not be copyrighted individually, their “choice, sequence and combination” may warrant copyright protection as an intellectual creation of the author (para. 66-67) under general copyright law.
- 48 However, the functional character of APIs, being even stronger than with computer programs in general, would very often place them well below the originality threshold,²⁹ and the general support in favour of interoperability expressed by the Software Directive could possibly present a further counterargument. As a very tangible sign of that support, art. 6(1) of the Software Directive states that reverse engineering by way of decompilation of program object code is permitted solely when it is the only way to obtain the information necessary to achieve interoperability with other programs. Thus, it could be held that copyright protection should be denied to the SSO of an API in case this hampers interoperability. A further question would be, however, if this applies *also* in the case of interfaces reaching a level of imperfect interoperability, as it happened with the reproduction of the SSO of the 37 Java APIs. In other words, the question could be raised whether the type of imperfect compatibility between the Java and the Android platforms achieved via the replication of the SSO of Java APIs would still be in line with the Software Directive’s strong *pro*-interoperability stance. Of course, both *SAS v. WPL* and *Oracle v. Google* can be seen as promoting *horizontal* interoperability and, by that, increasing *interplatform* competition. This effect is apparent especially in the EU case. In fact, SAS users are no longer discouraged from mo-

ving to a new, better (i.e. cheaper and/or more innovative) platform. Thanks to WPL's efforts, they can easily migrate their data and scripts to a competing, alternative environment. In the *Oracle v. Google* case, however, the result of Google's interoperability efforts is quite different. Applications written in Java can run on the competing Android platform only to the extent in which they call the functionalities of the 37 Java API packages whose SSO were replicated by Google. It follows that the level of interplatform competition between Android and Java made possible by Google's interoperability efforts is much less intense than that between the SAS and WPL platforms. On the other hand, it would not seem advisable to burden the copyright analysis of these issues with the difficult assessment of the suitable, pro-competitive level of interoperability under the circumstances of each specific case.



* An earlier version of this paper was presented at the Conference of the Fellows of the OpenForum Academy, held in Brussels, 24th September 2012. The author can be reached at simonetta.vezzoso@unitn.it

- 1 See Urs Gasser and John Palfrey, "Mapping Cloud Interoperability in the Globalized Economy: Theory and Observations from Practice," National Centres of Competence in Research, Working Paper No 2012/25, 9.
- 2 Thus, for instance, the relationships between Twitter and its ecology of app developers have started becoming rather tense as of August 2012, see "Twitter horrifies third-party developers with warnings of cutoffs," *The Guardian* 17.08.2012.
- 3 Ex-post, antitrust policy plays a very important role, in particular by prohibiting what in EU competition policy terms would be called abuses of dominant position as to Art. 102 TFEU. Thus, in the well-known EU Microsoft five-year investigation, the issue of software interoperability had been central to the decision to impose a €497 fine and the obligation to disclose interoperability information, see Commission Decision of 24.03.2004, Case COMP/C-3/37.792.
- 4 See Proposal for a Council Directive on the legal protection of those computer programs, COM(88) 816 final, OJ 12.4.89, No C 91/4, at § 3.7, available at [http://aei.pitt.edu/13138/1/COM_\(88\)_816_final.pdf](http://aei.pitt.edu/13138/1/COM_(88)_816_final.pdf). With regard to the details of the three-year process leading to the adoption of the Software Directive see Alan K. Palmer, Thomas C. Vinje "The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development" (1992), 2 *DUKE JOURNAL OF COMPARATIVE AND INTERNATIONAL LAW*, 65.
- 5 Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs, replaced by European Parliament and Council Directive 2009/24/EC of 23 April 2009 (consolidated version, partial renumbering). The CJEU in *SAS v. WPL* refers to the text of the 1991 Directive.
- 6 Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, "A Manifesto on the Legal Protection of Computer Programs" (1994), 94 *COLUM. L. REV.* 2308, 2320.
- 7 *SAS Institute Inc v. World Programming Ltd* [2010] EWHC 1829 (Ch).
- 8 While decompilation, i.e. the conversion of the machine-readable object code back to source code, involves a significant manipulation of the original software, black box analysis is much less burdensome. Solely the source code of the target program (the exterior of the "box") is put to intensive observation by executing a series of tests.
- 9 Opinion of AG Bot, delivered on 29 November 2011, Case C-406/10, *SAS Institute, Inc v. World Programming Ltd*.
- 10 [2010] ECR I 3971.
- 11 See the 1989 Proposal (n. 4), para. 1.1 ("a set of instructions the purpose of which is to cause an information processing device, a computer, to perform its functions").
- 12 Pamela Samuelson, "Why Copyright Law Excludes Systems and Processes from the Scope of Its Protection," (2007) 85 *Texas Law Review*, 1921, 1934.
- 13 TRIPs art. 9(2); WCT art. 2.
- 14 *Supra* (n. 4).
- 15 Pamela Samuelson, Thomas Vinje, And William Cornish, "Does Copyright Protection Under the EU Software Directive Extend to Program Behaviour, Languages and Interfaces?" (2011) 34 *European Intellectual Property Review*, 156, 163.
- 16 Cfr. Sally Weston "Software Interfaces – Stuck in the Middle: The Relationship Between the Law and Software Interfaces in Regulating and Encouraging Interoperability" (2012) *International Review of Intellectual Property and Competition Law* (IIC), 427, 442. Contrary to AG Bot's restrictive approach, the question should be raised whether it would be advisable to revise the Software Directive in order to significantly broaden the decompilation provision, see Andreas Wiebe, "Interoperabilität von Software" (2011) 2 *Journal of Intellectual Property, Information Technology and E-Commerce Law* (jipitec), 89, 95, available at <http://www.jipitec.eu/issues/jipitec-2-2-2011/3081>.
- 17 Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonization of certain aspects of copyright and related rights in the information society (OJ 2001 L 167, p. 10).
- 18 Jochen Marly, "Der Schutzgegenstand des urheberrechtlichen Software-schutzes" (2012) *GRUR*, 773, 778.
- 19 The Oracle/Sun merger raised in Europe some serious competition concerns, eventually cleared by the Commission, see Carl-Christian Bühr et. al., "Oracle/Sun Microsystems: The Challenge of Reviewing a Merger Involving Open Source Software" (2011) *Competition Policy Newsletter*, 20; Simonetta Vezzoso, "Open Source and Merger Policy – Insights from the EU Oracle/Sun Decision (2011) *International Review of Intellectual Property and Competition Law* (IIC), 344. None of those concerns, however, explicitly referred to Oracle's acquisition of the Java development environment created by Sun.
- 20 Final Judgment. Throughout this article reference will be made to Judge Alsup's Order re copyrightability of certain replicated elements of the Java application programming interface, available at <http://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1202/0.pdf?ts=1338544292>.
- 21 The Court of Appeals for the Third Circuit, which, analogizing to the plot of a book, originally stated that the SSO of computer programs are protected as long as they do not qualify as an idea, adopted a very broad standard, declaring that only the basic purpose of the program is unprotected, see *Whelan Assocs., Inc. v. Jaslow Dental Laboratory, Inc.* 797 F.2d 1222 (3d Cir. 1986). The broad (pro-copyright) Whelan standard, however, was mostly disregarded by subsequent court decisions, see Josef Drexel, "What is Protected in a Computer Program?" *Studies in Intellectual Property and Copyright Law*, Vol. 15. New York: VCH Publishers Inc, 1994, 20 et seq. Still, the general trend in the U.S. has been to endorse some form of copyright protection of SSO.
- 22 For instance, the Oracle chief Java architect testified during the trial that it took him and his team of engineers two years to develop the *java.nio* package.

- 23 *Satava v. Lowry*, 323 F.3d 805, 812 n.5 (9th Cir. 2003).
- 24 Christian Heinze, “Software als Schutzgegenstand des Europäischen Urheberrechts” (2011) 2 *Journal of Intellectual Property, Information Technology and E-Commerce Law* (jipitec), 97, 101, available at <http://www.jipitec.eu/issues/jipitec-2-2-2011/3082>.
- 25 *Supra* (n.4), para. 3.13.
- 26 Heinze (n. 24), at 101.
- 27 *Case C-5/08, Infopaq Int’l A/S v. Danske Dagblades Forening* [2009] ECR I-6569, paras. 37-51. See for a U.S./EU comparison on this issue Connor Moran “How Much is too Much? Copyright Protection of Short Portions of Text in the United States and European Union after *Infopaq International A/S v. Danske Dagblades*” (2011) 6 *WASH. J.L. TECH. & ARTS* 247, available at <https://digital.lib.washington.edu/dspace-law/handle/1773.1/563>.
- 28 Heinze (n. 24), at 101.
- 29 Marly (n. 18), at 779; Ashwin Van Rooijen, “The Software Interface between Copyright and Competition Law: A Legal Analysis of Interoperability in Computer Programs”, *Kluwer Law International*, 2010, at 79.