International Review of Intellectual Property and Competition Law

2012

Software interfaces - stuck in the middle: the relationship between the law and software interfaces in regulating and encouraging interoperability

Sally Weston

Subject: Information technology. **Other related subjects:** Competition law. Intellectual property

Keywords: Competition law; Compulsory licensing; Copyright; EU law; Interoperability; United States; User interfaces

Legislation: Treaty on the Functioning of the European Union art.102

Directive 2009/24 on the legal protection of computer programs

Cases: Microsoft Corp v Commission of the European Communities (T-201/04) [2007] E.C.R. II-3601 (CFI)

SAS Institute Inc v World Programming Ltd (C-406/10) [2012] E.C.D.R. 1 (AGO)

*IIC 427 Abstract

Interoperability of software programs is associated with competition and innovation. This has resulted in exceptions to copyright protection for software interfaces and intervention by competition authorities. Despite this, the hidden and unreadable nature of machine code and the limitations of reverse engineering have given a much stronger protection than is normally associated with copyright. This article reviews the theoretical and empirical justifications for interoperability. Then, against the backdrop of the prevailing uncertainty pending the decision of the ECJ in the case of *SAS Institute Inc v. World Programming Ltd,* the author analyses the development of the law on the status of software interfaces in Europe and the USA. The effectiveness of compulsory disclosure of interoperability information under the "exceptional circumstances" test is considered but the shortcomings of this approach are identified. The possibility of relaxing the restriction on dissemination of interface information is proposed.

I. Introduction

Two main legal devices exist to enable interoperability between software packages: the Software Directive[1] and Art. 102 TFEU. The Software Directive provides for software to be protected by literary copyright but permits reverse engineering for the purposes of interoperability. The protection afforded to interfaces has been considered most recently by Advocate General Bot in his Opinion[2] delivered following the referral to the ECJ of the case of *SAS Institute Inc v. World Programming Ltd.* [3] Although the Directive applies to all computer programs, there are practical limitations as the Directive does not mandate the disclosure of interface details, and upgrades can defeat reverse engineering. The second relief is that afforded under Art. 102 TFEU. However this is only available where the supplier is dominant and the refusal to disclose interface information amounts to an abuse. While this does mandate disclosure, it is an ex-post remedy and runs into arguments that the compulsory licensing of intellectual property rights (IPRs) can reduce incentives to innovate.

IIC 428 There are serious shortcomings in both approaches, and neither ensures a workable solution. The order for disclosure of interface information in *Microsoft v. Commission* [4] and recent merger cases assists those particular markets. The opinion of the Advocate General in *SAS Institute* is welcome, but it did not address all of the relevant aspects, and it does not help to encourage or mandate the disclosure of the interface information. While interface information remains effectively hidden, proprietary software companies benefit from a stronger protection than would normally be provided by copyright. Although reverse engineering, the voluntary disclosure of interface information, or the use of standards can and do assist, lack of interoperability and lock-in exists in many important areas, increasing cost and reducing competition.

II. Reverse Engineering

The **Software** Directive gives literary copyright protection to “the expression in any form of a computer program”[5] while recognising that the “function of a computer program is to communicate and work together with other components of a computer system and with users”.[6] For this to happen a particular piece of **software** must interoperate with other pieces of **software**. One way of achieving this is to “read” the interface of the **software**; however, the **software** user cannot *see* the rules and codes of the **software** in the same way that the reader of a book can *see* the text of the book. To gain this information he needs to take steps which would otherwise be reserved to the rightholder.

The Directive has explicit exceptions to enable this to happen. These exceptions do not require the rightholder's consent and cannot be contracted out of. The exceptions include the right to make a back-up copy and “to observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program”.[7] This latter exception is known as “black box” analysis and is not limited to interoperability. When black box analysis is insufficient to achieve interoperability, reverse engineering is permitted if, in order to achieve interoperability of an independently created computer program with other programs, it is necessary to reproduce the code and translate its form.[8] This converts the machine readable object code, which is the version normally supplied to the public, back into a higher level language, resembling the original source code which can be read by humans.

This exception is subject to certain conditions which emphasise that the exception can only be used to achieve interoperability, and not to create a ***IIC 429*** computer program substantially similar in its expression,[9] or for any other act which infringes copyright. This exception allows for the decompilation of the object code, in other words the user is allowed to look at and understand the basic building blocks of the program.

The social welfare benefits of reverse engineering have been described as complicated and ambiguous.[10] Reverse engineering provisions did not appear in the first draft of the **Software** Directive and were only inserted after a battle between various factions of the **software** industry and user representatives.[11] Unfortunately the hard won provisions on reverse engineering have severe practical limitations, and they do not always give a complete answer to the problem of interoperability.[12]

In its decision on *Microsoft,* the Commission found as a matter of fact that reverse engineering would not constitute a viable solution for companies wanting to compete in the work group server operating system market.[13] The volume of **interfaces** that would have to be reverse engineered in a program as large as Windows would require considerable effort with uncertain prospect of success. The viability of products developed using reverse engineering depends on the rightholder not altering its **software** so that it is no longer compatible with the new **software** developed by reverse engineering. Such alterations frequently occur when upgrades are issued. Reverse engineering is an inherently unstable basis for a business model. The Commission's decision referred to **software** developed by reverse engineering by ***IIC 430*** the Samba group. More than two years after Windows 2000 had appeared on the market, the Samba **software** still had severe shortcomings.[14] Also, **software** developed by Novell to interface with Windows NT was not compatible with Windows 2000. Microsoft would not release vital interface information to Novell, but used the lack of interoperability to discourage customers from using Novell's product.[15] In the more recent decision on the merger of Intel Corporation and McAfee Inc.[16] the market investigation revealed that most respondents considered that reverse engineering Intel's CPUs would take months if not years, be prohibitively expensive, and still be incomplete and vulnerable to subsequent changes to the CPU.[17]

During the formation of the Directive there was recognition that while it is technically possible to decompile a program, doing so is lengthy, costly and inefficient.[18] However at the time it was said that “the problem of access to information may have to be addressed by other means which are outside the scope of the Directive”.[19] One efficient solution is for the parties to voluntary disclose information on agreed terms.

III. Voluntary Disclosure and Standards

There may be good reasons why firms, even competitors in the same market, disclose interoperability information to each other when they would not share other IPRs. Indeed the voluntary exchange of this information is common practice. In network markets, suppliers often make interface information available to encourage adoption of their system. This practice was recognised in *Microsoft,* and the fact that Microsoft had supplied the information for the purpose of encouraging the adoption of its work group operating system was considered of relevance. The need for interoperability has spawned new companies whose *raisons d'#tre* are to provide **software** which allows **software** from competing suppliers to interoperate. Voluntary disclosure is however far from universal, always at the gift of the rightholder and at present is not an adequate solution.

Furthermore, standards do not provide a complete answer, particularly for complex **software**. One example of this is 3D Computer Aided Design (3D CAD) **software**. In this market, standards for data interchange do exist, including STEP, probably the most promising solution. STEP is an ISO standard for the exchange of product model data which has developed ***IIC 431** independently of the **software** suppliers. Standards such as STEP do allow for some data exchange between users and can reduce the legacy problem, but the information exchanged is frequently incomplete. For example the STEP files frequently open in non-native CAD applications as surfaces rather than solids, and vital history tree information is often missing.[20] The lack of adequate standards has encouraged the development of **software** companies specialising in **software** to provide interoperability, but their success in providing a technical solution or being affordable to all users is mixed. Less than 33% of engineering companies surveyed used a third-party translator, and of those only 45% indicated that they get the results they want with these applications better than 75% of the time.[21]

IV. Compulsory Licensing

The shortcomings in the Directive could be overcome by requiring the supplier to disclose the interface information. This was the nature of the order in *Microsoft* under Art. 102 TFEU (then Art. 82 EU) in response to the finding by the ECJ that Microsoft had abused its dominant position by failing to disclose interface information. The wisdom to date has been that this would involve a compulsory licence of IPRs which could be detrimental to the incentive to innovate.[22]

Competition law and IPRs have the same goals - to maximise both allocative efficiency (cheaper products with less resources) and dynamic efficiency (superior products) - but there is conflict in the way they operate.

There is a view that markets are superior to any form of intervention including competition law,[23] but IPRs are themselves inherently a form of intervention intended to encourage innovation and subsequent competition. The IPR regime is not ideal as neither patents nor copyright are a good fit for **software**. In the case of copyright, the term of protection bears no resemblance to the useful life of **software**. Copyright is a “weak” protection intended for literary and artistic expression that is normally exposed to ***IIC 432** numerous competing expressions in an imperfect market.[24] **Software** is functional in nature and most rights protected by copyright (for example the text of a book) are available to the world, but **software** codes are not visible to the user of the **software**. Reverse engineering is required to make the **software** code available. As the imperfect intervention of IPR already exists, consideration should be given as to whether market forces can be relied on to give optimum conditions, or whether further intervention, for example in the form of competition law, is necessary to prevent harm to consumers.

The use of Art. 102 to regulate the disclosure of interface information has the virtue that its limitations are well recognised when it comes to encouraging innovation. In other ways, however, its use is undesirable. It is ex post and prone to error. The limits of competition law were acknowledged in the seminal article of similar name by Frank Easterbrook.[25] The shortcomings of the attempts by competition law to deliver an economic ideal by participants lacking essential information were recognised. Easterbrook cautioned against over enforcement of competition law as the courts' false positive errors, which become law and are followed in later cases, are likely to take longer to correct than false negatives. “… judicial errors that tolerate baleful practices are self-correcting, while erroneous condemnations are not”.[26]

Accordingly any approach that encourages courts to reduce the protection afforded by IPRs, and the encouragement to innovation they provide, should be carefully scrutinised.

V. Creative Destruction

Interface information provides interoperability. Without interoperability markets are likely to exhibit network effects.[27] Innovation, if it happens, may take the form of the “gale of creative destruction”[28] rather than evolution.

Markets with products that have incompatible standards tend, after a period of intense competition, to have one single firm emerge dominant. As one *IIC 433 firm entices more developers and consumers to its standard the market may "tip".[29] It is argued that technological innovation in the new economy markets is so rapid that no market leader, even with strong network effects, can defend itself against new market entrants with "killer applications" meaning that "serial monopolies" are the norm with competition "for the market" rather than "within the market".[30] It is however possible that a firm may win the de facto standards battle, not on merit but with the help of a few tactical antitrust violations, and then hold the market for a long time.[31]

This "winner-takes-all" model means that entering the market is very risky. The new entrant has to secure strong economies of scale and network effects to enable it to "leapfrog" the former market leader. This makes investing in possible new entrants very risky and deters investment in potentially better products. The serial monopoly hypothesis claims that an innovator needs a period of monopoly in order to recoup its investment in innovation.[32] Strong IPRs are needed to attract investment in innovation where there are network barriers to entry.

The model of serial monopoly with strong IPRs and little competition law interference has attractions, as false positive errors are avoided. However, dominant companies can use lack of interoperability to leverage their position and attempt strategic foreclosure. In the Browser War[33] Microsoft leveraged its position in operating systems to ensure OEMs preinstalled Internet Explorer in a prescribed manner. Microsoft protected its share of the browser market, but a more significant effect was that by damaging Netscape Navigator, which could run on several operating systems, Microsoft was able to protect its position as the dominant supplier of operating systems. The pace of competition law meant that judicial relief came too late for Netscape.

Incumbents who have essential IPRs may attempt to steer innovation and the evolution of technologies, partly by innovating faster themselves (positive effects) but also by attempting to thwart innovation by potential competitors who may challenge their dominant position (negative effects).[34] IPRs in interface information belonging to dominant companies can be used to *IIC 434 prevent the emergence of superior technology which is not compliant with the de facto industry standard. This allows the dominant company to ratchet up its IPR protection.

The benefit of IPRs is that the owner can exclude others from using the protected subject-matter, but this exclusivity is not an exemption from competition. It is only an instrument that compels the proprietor's competitors to compete by substitution as opposed to imitation.

VI. Interoperability and Innovation

Interface information enables **software** to interoperate with other suppliers' **software** and platforms. It appears that there is no clear empirical evidence of a link between interoperability, competition and innovation, although intuitively it would seem that interoperability would create an expansion in use, enabling competition and encouraging innovation.[35] The innovations thus stimulated would be more likely to be of the "follow on" type rather than "breakthroughs".[36] Studies have found anecdotal evidence sufficient to support the claim of a link between interoperability and innovation.[37] It is certain that lack of interoperability causes expense and wastage. The National Institute of Standards & Technology estimated that imperfect interoperability cost the US automotive supply chain at least $1 billion per year in 1999.[38] Incompatibility between two versions of Dassault Systemes' CATIA 3D CAD **software** resulted in a $6 billion loss for Airbus in the design, and delayed the delivery of the A380.

Consumers as well as industrial users may be affected by interoperability issues. Once consumers have bought a platform or **software** they may be unable to move their data. They are "locked in",[39] and if they have not chosen the emergent de facto market standard they are faced with the additional costs of changing to the market standard and may lose the use of their expensively acquired data. This dilemma is apparent in the new market for e-readers. At least two of the e-readers are closed systems. Users who *IIC 435 build up a library do not know whether they can transfer it to another platform. A study commissioned by the Book Industry Study Group identified the number one complaint among consumers about e-readers as "certain e-books [are] specific to certain e-readers".[40]

The law has recognised the need for exemptions from IPRs to permit **interfaces** between objects in more traditional industries. One early example is the "must fit" or "must match" exemption to the design right features of a product. These features are ones which must be exactly reproduced in shape and size so that the product can be connected to another product, enabling either product to perform its function.[41] This means manufacturers may be prevented from stopping competitors from making spare parts for their products. The **Software** Directive is a more recent manifestation of the law trying to balance the needs of interoperability with those of IPR holders.

VII. Interfaces and Copyright

Although the **Software** Directive grants copyright protection to the expression in any form of a computer program, "ideas and principles which underlie any element of a computer program, including those which underlie its *interface,* are not protected."[42] This is stated "for the avoidance of doubt" in the recitals and in the operative part of the Directive.[43] The Directive also acknowledges that the function of a computer program is to communicate and work with other components of a computer system and with users.[44] To achieve this interconnection and interaction it is required "to permit all elements of **software** and hardware to work with other **software** and hardware and with users in *all the ways in which they are intended to function.* The parts of the *program* which provide for such interconnection and interaction are generally known as '***interfaces***'*."* [45] [emphasis added].

The functional interconnection and interaction is generally known as "interoperability" which is defined as the "ability to exchange information and mutually to use the information which has been exchanged".[46]

Interfaces comprise not only the code that implements them but also the ideas, rules or principles in the specification of the interface. The specification can be used by a programmer to create an independent implementation ***IIC 436** of the interface which uses different code. As the specification amounts to ideas and principles it is not subject to copyright. The phrase "where the specification of **interfaces** constitutes ideas and principles which underlie the program, those ideas and principles are not copyrightable subject matter" appeared in the original proposal but was removed from the final Directive. It was criticised for being so obvious that it would introduce the suggestion that this may not always be so.[47] It was also objected to by members of the **software** industry who claimed that excluding specifications from copyright protection would facilitate "piracy".[48]

We therefore have a scenario where at least some aspects of **interfaces** do not have copyright protection, but even so the **interfaces** are not readable without reverse engineering.

Article 9(2) of TRIPS and Art. 2 of the WCT state that copyright protection shall extend to expressions and not to "ideas, procedures, methods of operation or mathematical concepts as such". Neither domestic UK law nor the **Software** Directive expressly contain the exclusions that copyright does not extend to "… procedure, method of operation, or mathematical concepts as such". Nevertheless domestic and European law must now be interpreted in conformity with TRIPS and Art. 2 of the WCT. [49]

By contrast Sec. 102(b) of Title 17 of the United States Code was amended in 1980, in line with TRIPS, to deny copyright protection to any "idea, procedure, process, system, method of operation, concept, principle, or discovery." [50] At the time the **Software** Directive was being debated and introduced in Europe, the US case of *Whelan v. Jaslow* [51] gave a conservative interpretation that only the purpose or function was the idea and everything else was the expression. This meant that **interfaces** could be swept up in the broad concept of structure, sequence and organisation, as being copyright protected. In 1992, after the **Software** Directive had been adopted in Europe, the US position shifted. The case of *Computer Associates v. Altai* [52] applied "the abstraction-filtration-comparison test" [53] and filtered out in the second stage those facets that are dictated by external constraints, such as compatibility with other programs, as not copyright protected. In the same year ***IIC 437** reverse engineering was considered fair use, for if disassembly was per se unfair this would give a de facto monopoly over the functional aspects.[54] A monopoly over ideas or function must satisfy the more stringent test imposed by patent law.[55] This case is important not only because it condoned reverse engineering but also because it described the interface information as functional aspects.[56]

In *Lotus v. Borland,* [57] Lotus' menu command hierarchy, including macros, were considered to be methods of operation and excluded from copyright protection under Sec. 102. Borland wanted to emulate the Lotus **software**, not by copying the underlying code, but by copying the Lotus menu command hierarchy such as the "copy" and "print" commands. These commands explained and presented functional capabilities to the user and were the method by which the program was operated and controlled. They were essential to operation and the court considered it was not necessary to determine whether they could have been designed differently. Compatibility strengthened the argument that the menu command hierarchy was a “method of operation” as otherwise, in order to use other software, the user would have to learn many different operating methods. The court found this notion “absurd”.[58] That there are different ways to operate computer programs, and different ways to arrange hierarchically command terms, does not make the actual method chosen copyrightable. It functions as a method of operating the computer and is uncopyrightable. “The “expressive” choices of what to name the command terms and how to arrange them did not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter”.[59]

The US law has applied the methods of operation exclusion to some forms of interface. Thanks to TRIPS and the WIPO Copyright Treaty, the methods of operation exclusion will now apply to domestic and European law.

Against this backdrop there have now been a number of cases in Europe where software has been consciously emulated so that the user interface bears similarities to existing software. These cases have provided an interpretation of the copyright protection afforded to user interfaces by the Software Directive.

**IIC 438* VIII. Navatatre

In *Navataire Inc v. easyJet Airline Co Ltd.,* [60] Pumfrey J considered VT100 screens were literary in character. However they were “ideas which underlie its interface” in the sense used in Art. 1(2) of the Directive: “they provide the static framework for the display of the dynamic data which it is the task of the software to produce”.[61] By contrast graphical user interface (“GUI”) screens were artistic and outside the scope of the Software Directive.

In *Navataire* the source code had not been copied and there was no reverse engineering, but the new system was substantially indistinguishable from the original system in respect of its “user interface”. In addition to the question of whether copyright subsisted in the interface Pumfrey J also considered whether the copying of commands amounted to a substantial proportion, and whether copying the “business logic” aka “non-textual copying”, or copying without access to the thing copied, directly or indirectly, infringed the copyright in the source code. Pumfrey rejected the claims saying:

If it is the policy of the Software Directive to exclude both computer languages and the underlying ideas of the interfaces from protection, then it should not be possible to circumvent these exclusions by seeking to identify some overall function or functions that it is the sole purpose of the interface to invoke and relying on those instead.[62]

IX. Bezpe#nostní

In the case of *Bezpe#nostní softwarová asociace* [63] the ECJ was asked to determine whether for the purposes of Art. 1(2) of the Software Directive the phrase “the expression in any form of a computer program”, includes the graphical user interface of the computer programme or part thereof.

Here again the interface in question is the graphical user interface (“GUI”) rather than a pure software interface. The Advocate General said the question was whether the GUI “which is the *result,* on screen, of a computer program constitutes an expression”[64] [emphasis added]. The GUI is not seen as part of the code but only as a result of the code. The GUI, also referred to as the “look and feel” enables communication and interaction between the user and the program.

The ECJ considered that the GUI interface did not constitute a form of expression of a computer program as the GUI does not enable the reproduc ****IIC 439*** tion of that computer program. It is only one element which allows the user to use the features of the program. The ECJ referred to the Advocate General's opinion that “the form of expression of a computer program must be protected from the moment when its reproduction would engender the reproduction of the computer program itself, thus enabling the computer to perform its task”.[65] Copying the code, including into another language, would cause the computer to perform its task. Presumably some forms of incomplete copying would also come within the Advocate General's test, so the code does not have to be reproduced exactly, provided the computer performs the task expected. The GUI would not however enable the reproduction of the program.

In *Bezpe#nostní* the ECJ confirmed that aspects of certain interfaces, GUIs in particular, are not expressions and are not copyright protected. The case does not directly address whether the code creating the interface is protected by copyright or not. It also does not consider other forms of interface, namely data formats, protocols and APIs although it can be inferred from the Advocate General's test that the idea/expression dichotomy will vary depending on the nature of the interface involved.

X. SAS Institute Inc v. World Programming Ltd

The case of *SAS Institute Inc v. World Programming Ltd* [66] was referred to the ECJ on a number of points and Advocate General Bot's decision was delivered on 29 November 2011.

The case illustrates the problems that lack of interoperability poses for users and competitors. SAS Institute Inc ("SAS") is a major supplier of sophisticated analysis **software** using its own SAS language. Customers may have written thousands of application programs in the SAS language and while "there are other suppliers of analytical **software** which compete with SAS Institute, a customer who wanted to change over to another supplier's **software** would be faced with re-writing its existing application programs in a different language".[67] Customers were locked-in to the SAS **software** and had to continue to buy an annual licence. World Programming ("WPL") sought to provide an alternative, cheaper[68] **software** to enable users who had programs written in the SAS language to have a choice of **software**. WPL sought to emulate much of the functionality, to ensure the same inputs would produce the same outputs, and to ensure customer's application programs ran in the same manner on WPS and SAS components.

IIC 440 WPL did not have access to the source code nor did it decompile the **software**.[69] The source code for the WPL **software** was mainly written using the SAS manuals and observing the operation of a Learning Edition of the SAS **software**. The method WPL used to create the interface was characterised by observation of the format of data files to enable WPL to write source code which read and wrote data files in the same format.[70] SAS claimed that WPL had infringed the copyright in its manuals, and thereby indirectly the copyright in the programs, and breached the terms of the licence of the Learning Edition of the SAS **software**. The facts are similar to *Navitaire* but involve **interfaces** between the **software** and data files rather than the user interface.

Many provisions of the **Software** Directive, as transposed into English Law, were analysed by Arnold J in the High Court decision. Aspects of the SAS **software** were considered to be a language or aspects of functionality and therefore excluded from copyright protection[71] .

Arnold J then considered whether **interfaces** are protected by copyright in a computer program. He was faced with the situation where WPL had not reverse engineered the interface and at no time had they been able to see and copy the source code. In these respects the facts were similar to *Navitaire.* In *SAS Institute* the interface information had been obtained by examining the SAS System in operation to work out enough of the format of the SAS7BDAT data file to write a new source code which reads and writes data files in that format".[72] The element of the interface that was in question, arguably, was akin to the ideas, rules or principles by which the interface was specified. Arnold J recognised that methods of operation were also excluded from copyright.[73]

The original proposal for the Directive, referred to by counsel for WPL, clearly stated that these rules or principles by which **interfaces** are specified are not copyright protected. If a specification is used to write new code to achieve interoperability then that does not infringe copyright. But what of the code that actually constitutes the interface? This is the expression, and would appear still to be copyright protected. The original proposal says competitors are free to build on the identical idea (the specification) but may not use the same expression as that of protected programs (the code). It also proposes that similarities in the code which implemented the ideas, rules or principles due to the inevitability of certain forms of expression, where ***IIC 441*** constraints of the interface are such that different implementation is impossible, will not infringe copyright as the idea and expression are said to merge.[74]

Arnold J, agreeing with Pumfrey J in *Navitaire,* concluded that **interfaces** were not protected by copyright. He thought the legislative history supported this and the inclusion of reverse engineering in the later version of the Directive was not counter to that interpretation. The purpose was to entitle third parties to obtain information about **interfaces** in one or more ways. This he concluded meant that once the information is obtained it was intended that "competitors would be free to copy the interface anyway".[75]

The question is: what constitutes copying the interface? If it is using the specification to produce new code, then that would equate to using the ideas and principles which underlie the interface. This is in line with the wording of the Directive. Copying the code itself is something else. The only reference to this is in the Directive's original proposal where it speaks of the form of expression being constrained where the idea and expression may merge.[76] The courts should however consider whether the interfaces amount to methods of operation and are therefore exempted from protection.

Arnold J then considered what aspects of the SAS software amounted to an interface. The syntax of the SAS Language was part of the programming language rather than an interface, but as such is still unprotected by copyright. The SAS data file formats were considered “precisely the kind of information which is required by third parties in order to access data stored in those formats for the purposes of interoperability”[77] and thus were interfaces. WPS could read and write files in the SAS data file format but this did not amount to an infringement of the copyrights in the SAS components. There is no evidence that WPS reproduced a substantial, or indeed any part of the SAS source code. Instead they had examined the system in operation and worked out enough of the format of SAS7BDAT data files to write their own code.

When Advocate General Bot referred to his own reasoning in *Bezpe#nostní* about what constitutes expression, he said that the protection of a computer program is not confined to the literal elements, to the source code and object code, but extends to any other element expressing the creativity of the author.[78] He acknowledged the impact of the WCT and that not only are ideas excluded from copyright but also procedures, methods of operation and mathematical concepts.[79] He then explained that he considered the ***IIC 442** functionalities and language of a computer program are not capable, as such, of being protected by copyright. To determine whether copyright exists, account should be taken not of the time and work or level of skill, but the degree of originality.[80] He said that the language is a functional element lacking any originality and drew a comparison with the language used by the author of a novel. This does not stop code written in the language from copyright protection as this would amount to the expression.[81] It will be for the national courts to examine whether reproducing functionalities has also reproduced “a substantial part” of the elements of components “which are the expression of the intellectual creation of the author of those components”.[82] The opinion supported the distinction between ideas and expression in respect of functionality and language.

The opinion is less helpful on the question directly concerning the program interface and whether reading and writing files in the same format is making use of the interface's idea, principle, methods of operation or its expression. Advocate General Bot rephrases the question to ask whether copyright has been infringed by “deciphering” the format of the SAS data files to write a new source code. Arguably deciphering the format can be done either by observation or by some form of translation. Only if translation is involved does it amount to reverse engineering under the Directive. As stated earlier, it was found as a matter of fact that reverse engineering did not take place. WPL had observed the format files. They had not translated or altered SAS' code in the way that would amount to reverse engineering, as described in Art. 4(b). The Advocate General's Opinion however focuses on Art. 6 of the Directive which deals with reverse engineering. It says Art. 6 should be interpreted strictly, decompilation should be an exceptional act, and the licensee will have to demonstrate the “absolute necessity” of its actions.[83] This language is even more restrictive than the language of the Directive and may not help the cause of interoperability.

Arnold J stated that “ interfaces as described in recital [15] of the Software Directive are not protected by the copyright in a computer program”. The Advocate General's opinion states that the Directive “does not exclude interfaces from copyright protection”, merely the ideas and principles underlying the interface. There may be less difference in these statements than meets the eye but the position should be clarified. Interfaces are not per se outside the protection of copyright. Source code and machine code interfaces may be copyright protected but there are certain aspects, such as specifications and protocols which are not expressions but ideas and principles and thus not copyright protected. Consideration should also be given by the ECJ, when deciding *SAS Institute,* as to whether the data formats and other interfaces amount to a method of operation and are exempted from protection.

***IIC 443** *XI. Microsoft*

Microsoft's European case occurred before the *Bezpe#nostní* ruling or the opinion in *SAS Institute,* but it is now considered to illustrate the relationship between Art. 102 and the Software Directive and how this may be influenced by the status of computer program interfaces. Unfortunately a definitive picture of the relationship cannot be drawn, as neither the Commission nor the General Court determined what IPRs existed in the information Microsoft was ordered to disclose. They said they did not need to, as Microsoft had abused its dominant position and met the exceptional circumstances test which justified the disclosure. The Court sidestepped the issue of whether the interface information was copyright protected. While noting that the contested decision "does not take a position as to whether Microsoft's [IPRs] are affected or not"[84] it proceeded on the presumption that the protocols in question, or the specifications of those protocols, are covered by IPRs. Compulsory disclosure was justified by the abuse of a dominant position and the exceptional circumstances test following the cases of *Magill* [85] and *IMS Heath.* [86]

The decision in *Microsoft* has been criticised for being a false positive, penalising success in the absence of abuse, which will reduce incentives to innovate and dynamic competition.[87] There are also claims that the "exceptional circumstances" test was distorted by the interpretation of the "indispensability" requirement[88] and the expansions of the new product test in a secondary market to technical development.[89] These issues have been debated elsewhere and only certain aspects relating directly to the interface will be considered.

Microsoft claimed the **interfaces** were protected by patents, copyright and trade secrets. However the Commission said the considerations associated with patent protection did not justify the refusal, indeed Microsoft took ***IIC 444** some time to even identify a single patent.[90] The Commission was also not impressed with the trade secret argument as the protection afforded to trade secrets can be more limited than copyright or patent protections, and they exist as a result of a unilateral business decision dependant on its facts and the interests at stake. Here the value of the secret was not its innovative nature but the fact that it belonged to a dominant undertaking.[91] Also, subject to certain conditions, reverse engineering can legitimately disclose information for the purpose of interoperability, and this would defeat any attempt to protect interface information as trade secrets.[92]

When considering the question of copyright the Commission commented that while specifications may be copyrighted their implementation is not necessarily a copy but may result in new work. The order in *Microsoft* was for the disclosure of "complete and accurate specifications for the protocols". The use of the term "specifications" did not require Microsoft to disclose how it implemented the specifications in the source code. The term "protocol" related to "the rules of interconnection and interaction".[93] The wording of the order resembles the wording in recital 10 of the **Software** Directive defining **interfaces** as parts which provide "interconnection and interaction" to enable **software** to work "in all the ways they are intended to function".[94] Presumably an objective test would be applied to this later phrase but the meaning of the phrases has not been established, although in *Microsoft* two-way operability was required.

Microsoft argued that the information disclosed went further than was necessary for interoperability. The information would allow for "plug-replaceablity", "cloning" and argued that "the replication of the Global Catalog features of the Active Directory do not bear on interoperability".[95] The Commission refuted these arguments and set out details of why Microsoft's "one-way" interpretation of interoperability was incorrect.[96] The Court considered Microsoft had not cast doubt on these assertions.[97] It considered the concept of interoperability adopted in the order namely "the capacity for them to exchange information and to use that information mutually in order to allow each of those **software** products to function in all the ways envisaged" as consistent with the **Software** Directive.[98] Nevertheless Art. 82 (now Art. 102) was considered to rank higher than the Directive, and the level of interoperability ordered was necessary to enable competitors to ***IIC 445** remain viable on the market.[99] Microsoft's assertions that aspects of the information to be disclosed did not relate to interoperability and could amount to cloning were rejected. The ECJ dealt with many points including that Microsoft would not be required to disclose the source code and only give a general description of the algorithms, leaving it to its competitors to develop their own implementation of it.[100]

On the face of it the information Microsoft was ordered to disclose: "Complete and accurate specifications for the protocols", was interface information for interoperability and hence not copyright protected.[101] The main problem was that the information arguably went beyond interoperability into the "internal make up of Windows server operating systems" [102] which Microsoft claimed could compromise the internal integrity and security of the system. The argument was not accepted by the Commission and it is debatable whether a dominant undertaking should be entitled to cite security risks to justify non-disclosure of information when that is dependent on its own architectural design decisions.[103] Another issue was the requirement to provide an algorithms' "general description" which can amount to a most valuable but unprotected aspect of **software**[104] demonstrating the misfit of copyright protection for **software**.[105]

It has been pointed out that the only example of reduced interoperability present at the trial was that some users needed to log-on twice and that the fines and extra work imposed on Microsoft seem disproportionate.[106] On the other hand factors relevant to the abuse were that Microsoft had formerly disclosed interface information to enter, catch up and then dominate the market, before introducing new **software**, Windows 2000, and not disclosing equivalent information. The new **software** was built on industry standard access protocols and open source **software**, LDAP and Kerbos, that was not developed by Microsoft, but to which Microsoft had added private *IIC 446 unilateral extensions.[107] Microsoft's behaviour in that period appeared to be intended to protect and exploit its dominant position[108] in the PC operating systems market, whether from Netscape and Java or other competing platforms such as a rival server operating system.[109]

XII. Merger Cases

In recent merger cases the Commission appears to have accepted the position that a lack of interoperability has an adverse affect on competition, particularly in network markets.[110] In the decision on Cisco's acquisition of Tandberg,[111] the lack of interoperability was seen as a barrier to entry to the market for high-end video conferencing systems. The question in that merger was whether the market would impose interoperability or whether intervention was required. Perhaps unsurprisingly, the competitors favoured intervention. End customers, distributors and industry analysts however thought that the market would impose interoperability and that a standard would develop.[112] The market investigation revealed that network effects meant there was a strong case for interoperability. It confirmed that interoperability was the way forward for the industry, but the merged entity could have an increased incentive to strategically restrict interoperability with new entrants or less important competitors.[113] Cisco was required to divest the copyright and management of its TIP protocol to an industry body before the merger was approved.[114]

The acquisition of the security **software** vendor McAfee by Intel raised concerns as competing security **software** vendors would continue to need good interface information post-merger, information which is essential to *IIC 447 ensure that their **software** was not disadvantaged with regard to performance and power consumption, as that would significantly increase workload on the CPU and affect performance of the computer.[115] There was concern that after the acquisition Intel would lose the incentive to disclose information to competing security **software** vendors and this would foreclose the market. Given Intel's large market share, customers would not be in a position to exert pressure on Intel to restore interoperability and reverse engineering was not “commercially viable or technically feasible.”[116] This would have a significant adverse effect on the market, acting as a technical tie and foreclosing the market. Intel committed to give equal access to “Instruction, Interoperability and Optimization” information.[117] It is interesting that the Commission co-operated closely with the U.S. Federal Trade Commission throughout the review.[118]

XIII. Encouraging Interoperability - Concluding Remarks

The willingness to intervene to promote interoperability in *Microsoft* and more recent merger cases indicates that the Commission has been convinced by the competitive virtues of interoperability. Arguably, the outcome of *Microsoft* has strengthened the Commission's bargaining position in negotiations intended to maintain interoperability post merger. It will be interesting to see how the Commission and the European courts approach the next case brought under Art. 102 claiming that a lack of interoperability is harming competition. When it comes to organic growth will they look favourably on interoperability to encourage follow on innovation, or will they adopt the position that the new economy is a special case, and fares better from less intervention, thus allowing breakthrough innovation? If the Commission does look favourably on interoperability to promote competition and innovation it is hoped that the ECJ will share the positive view of interoperability and give a ruling in the *SAS Institute* case that will provide both certainty on the status of **interfaces**, and promote reverse engineering. A clear ruling on the status of **interfaces** would help to promote interoperability as part of the IPR regime, rather than relying on Art. 102 cases that only apply to dominant companies.

*IIC 448 Unfortunately, we do not know with certainty the extent to which IPRs were included in the information Microsoft was ordered to disclose. The General Court did not determine the issue and the extent of copyright in **interfaces** is also uncertain. It is arguable that the cases which developed the exceptional circumstances test, namely *Magill, IMS Health* and *Microsoft,* all concerned the attempt by the dominant entity to exceed the scope that the IPR regime was intended to protect.[119] Relatively weak IPRs allowed the use of a dominant position to leverage the position on, or block, a secondary market. In *Magill* the IPR was considered to be a subspecies of copyright, blocking the information rather than the expression. In *IMS Health* the copyrights were combined with a dominant industry standard which excluded competition. In *Microsoft* the level of IPR protection is unclear and some of Microsoft's IPRs may have originated from the public domain.[120]

The **Software** Directive permits black box analysis and reverse engineering, but it does not require suppliers to disclose interface information in any form. However, following *Microsoft,* once a supplier becomes dominant it may have to compile and disclose such information. We are faced with two levels of interoperability, in which the level of interoperability that can be required of dominant companies will be at a higher level than for other suppliers. This may be justified given the impact of network effects, but the very existence of the remedy implies that the interface information available under the **Software** Directive is inadequate for interoperability.[121] There will continue to be prospective suppliers unable to achieve interoperability for their **software**, and users locked in to incumbent suppliers. *SAS Institute* is only the tip of the iceberg. It could be argued that the cumulative effect of this does not affect the market and is inappropriate for a remedy under competition law, but it is a serious problem for the user and a cost to the economy and should be addressed as part of the IPR regime.

In *Microsoft* the court considered that competitors would not want to replicate exactly the same work-group server operating systems as Microsoft, but to offer something innovative and different. This supports the requirement under the “essential facilities test” for a secondary market with a new product or an advance in technical development. By contrast in *Navitaire* and *SAS Institute* the aim was to produce **software** with the same “look and feel”, to help users swap to the new supplier. It is ironic that a competitor has to establish a new product or technological advance to interoperate with a dominant company but can emulate the **software** of any other competitor.

IIC 449 The aim of literary copyright is to protect creativity rather than functionality, whereas the value of computer programs lies in their functionality and their algorithms, and not the more prosaic code used to achieve that functionality. There is a fundamental mismatch, and suppliers turn to protecting their interface information because of the lack of protection for parts of the program which truly merit protection.[122] This is exacerbated by the fact this is one of the rare occurrences when an IPR is not disclosed. Whatever the copyright status of the interface, the situation remains that the user cannot readily see the rules and codes of the **software**. Disclosure is the price usually paid for IPRs, but here there is no such requirement and this hampers competition by substitution or follow on technology. Network effects are reinforced as the boundaries of the network are established by poor interface information causing a lack of interoperability. The market has to rely on breakthrough technology, which, as seen in *Microsoft,* has to contend with sophisticated behaviour to protect **interfaces**.

Steps that could improve the situation, short of introducing a sui generis form of protection and exceptions, include a ruling by the ECJ in *SAS Institute* that clarifies the status of **interfaces** to give more certainty to the **software** industry. It would be incongruous for the ECJ to interpret the exception for **interfaces** restrictively under the **Software** Directive, while the Commission is encouraging interoperability through competition and merger decisions. The second step would be a change to the **Software** Directive to allow for the dissemination of interface information obtained by reverse engineering. At the moment the Directive prevents the disclosure of any interoperability information obtained by reverse engineering,[123] and this forces each supplier to repeat the painstaking decompilation for itself. Lifting this restriction would remove duplication of effort and allow firms to specialise in providing interoperability information to other vendors, or innovating in the knowledge that an interface is available.[124] This would create a market for interface information which could encourage, but not oblige, suppliers to make their own interface information available to ensure its quality, and could also bolster the use of standard **interfaces**. This market response will no doubt be resisted by many in the **software** industry with the same vehemence displayed when the **Software** Directive was introduced. Comparisons may be drawn with open source **software** where licence fees are lower, although their source code and not just interface information is available.

**IIC 450 XIV. Conclusion*

There is concern that compulsory licensing of IPR even in “exceptional circumstances” reduces the incentive to innovate. Arguably, in the main three cases of *Magill, IMS Health* and *Microsoft,* the disclosure concerned a weak IPR. When this is combined with the competitive and innovative benefits that are thought to flow from interoperability, it must cast a new light on the merits of the *Microsoft* case. The “exceptional circumstances” test is not however a good remedy for failure to disclose interoperability information. It is ex-post, complex and does not apply to most suppliers.

The **Software** Directive allows for black box analysis and reverse engineering, but the effectiveness of these exceptions is limited by the complex nature of the **software** and the ability to alter **interfaces** with upgrades. Copyright is not a good fit for **software**, and suppliers feel their valuable core areas are vulnerable, and compensate by attempting to overprotect **interfaces**.

The Advocate General in *SAS Institute* considered the dichotomy between ideas and expression of functionality and language, but not in respect of **interfaces**. It will be interesting to see what the ECJ says about **interfaces** and whether it will follow the Advocate General's comments on Art. 6, or consider **interfaces** more generally. Meanwhile, while some incumbent **software** suppliers voluntarily disclose interface information in proprietary **software** in response to commercial pressures, the lack of interoperability remains a cost to the economy and a burden on users. The ECJ has an opportunity to clarify the status of **interfaces** but a good resolution of the situation may still require amendment of the **Software** Directive.

Department of Law, Bournemouth University, email: sweston@bournemouth.ac.uk.

IIC 2012, 43(4), 427-450

---

[1] Council Directive 2009/24/EC on the legal protection of computer programs, (hereinafter “**Software** Directive” or “Directive” as case requires), [2009] OJ L111/16.

[2] Case C-406/10 *SAS Institute Inc. v. World Programming Ltd* [2011] ECR I -1, Opinion of AG Bot.

[3] *SAS Institute Inc v. World Programming Ltd* [2010] EWHC 1829 (Ch).

[4] Case *T-201/04 Microsoft v. Commission* 5 C.M.L.R. 11 [2007].

[5] **Software** Directive, *supra* note 1, at Art. 1(2).

[6] *Ibid.* at recital 10.

[7] *Ibid.* at Art. 5(3).

[8] *Ibid.* at Art. 6.

[9] Similar in expression prevents use of the same code but should not prevent the creation of a competing program. There has been no definite ruling on this point in Europe but in *Navitaire Inc v. easyJet Airline Co Ltd.,* (No. 3) EWHC 1725 (Ch) [2004], and *SAS Institute, supra* note 3, competing **software** emulating the “look and feel” was allowed under the **Software** Directive. *See also* the USA case of *Sega Enterprises Ltd. v. Accolade Inc.,* 977 F.2d 1540 (9th Cir. 1992), where Accolade used reverse engineering to create a competing video game, an application, and *also Sony Computer Equipment Inc. v. Connetix Corp.,* 203 F.3d 596 (9th Cir 2000), concerned operating **software**. These exercises in reverse engineering were permitted under the US “fair use” exception.

[10] *See* PAMELA SAMUELSON & SUZANNE SCOTCHMER, “The Law and Economics of Reverse Engineering”, 111 The Yale Law Journal 1575, 1621 (2002). Incentives to invest in platforms will be reduced as it allows unlicenced entry but this does not necessarily mean reverse engineering should be illegal. It probably increases the incentive to develop application **software**.

[11] Dominant American companies established a group called the **Software** Action Group for Europe (SAGE) which opposed reverse engineering while others promoted more interoperability through the European Committee for Interoperable Systems (ECIS), *see* C. MEYER & M. COLOMBE, “Interoperability still threatened by EC **Software** Directive: a status report”, 12(9) E.I.P.R. 325-329 (1990).

[12] For a full account of how reverse engineering is essential but complex and time consuming *see* ANDREW JOHNSON-LAIRD, “**Software** Reverse-Engineering in the Real World”, 19 U. Dayton L. Rev. 843 (1994).

[13] *Microsoft* (Case COMP/C-3/37.792 Microsoft), Commission Decision of 24 March 2004 relating to proceedings under Art. 82 EC, at para. 683 *et seq.*

[14] *Ibid.* at para 293 *et seq.*

[15] *Ibid.* at para. 686.

[16] Case COMP M.5984 Intel/MCAFEE [2004] OJ C 98-1.

[17]. *Ibid.* at para. 145.

[18]. *See also* PAMELA SAMUELSON & SUZANNE SCOTCHMER, *supra* at note 10, describing the considerable intellectual work and high costs involved in **software** reverse engineering and *see also* ANDREW JOHNSON-LAIRD, *supra* at note 12.

[19]. Proposal for a Council Directive on the legal protection of computer programs [1989] OJ C 91/4, at para. 3.14.

[20]. Despite enhancements ISO 1030, the STEP standard, did not achieve full “coverage of geometry and design intent” and remained “only a subset of what will be required for full practical translators” J. KIM ET AL, “Standardized data exchange of CAD models with design intent” 40 Computer Aided Design 760-777 (2008).

[21]. Longview Advisors, Inc. “Collaboration & Interoperability Market Report” 2008 http://www.proficiency.com/downloads/3DMarketReport2008.pdf.

[22]. *See* for example, ALAN DEVLIN, MICHEAL JACOBS & BRUNO PEIXOTO, “Success, Dominance and Interoperability”, 84, Ind. L.J.l 1156, (2009); *Daniel Spulber,* “Competition Policy and the Incentive to Innovate: The Dynamic Effect of *Microsoft v. Commission”* 25:2 Yale J. on Reg. 247 (2008); and THOMAS BARNETT, “Interoperability between Antitrust and Intellectual Property” 14 Geo. Mason L. Rev. 859 (2007).

[23]. The Chicago School considered the markets superior to any form of governmental, including judicial, intervention and judicial interventions generally have no coherent analytical basis. *See* GEORGE PRIEST, “The limits of antitrust and the Chicago School tradition”, 6(1) J.L. & E (2010).

[24]. DANIEL J. GIFFORD & ROBERT T. KURDRIE, “Antitrust approaches to dynamically competitive industries in the United States and the European Union”, 7(3) J.C.L. & E. 695-731, at 705-706 (2011). Gifford and Kurdie consider that the competition between differentiated products contemplated by copyright law does not work in network markets as the combination of copyright, trade secrets and the inaccessible code gives a much stronger protection than was probably intended by both the US and EC copyright initiatives.

[25]. FRANK H. EASTERBROOK, “The Limits of Antitrust”, 63 Tex. L. Rev. 1 (1984).

[26]. *Ibid.* at 3.

[27]. MICHAEL L. KATZ & CARL SHAPIRO, “Systems Competition and Network Effects”, 8 J. Econ. Persp. 94 (Spring 1994). The utility provided to each individual user of a specific platform or system increases with the total number of users, so the higher the number of users the higher the demand.

[28]. JOSEPH SCHUMPETER, “Capitalism, Socialism, and Democracy” 81 *et seq.* (George Allen, London 1976) first published 1942.

[29]. KATZ & SHAPIRAO, *supra* note 27, at 106 describing tipping as “the tendency of one system to pull away from its rival in popularity once it has gained an initial edge”.

[30]. DAN WIELSCH, “Competition policy for information platform technology”, 25(2) E.C.L.R. (2004).

[31]. WILLIAM PAGE, “Microsoft and the limits of antitrust”, 6(1) J.C.L. & E. 35 (2010); *see also* MARK LEMLEY, “Antitrust and the Internet Standardization Problem”, 28 Conn. L. Rev. 1041 (1996).

[32]. DAN WIELSCH, *supra* note 30.

[33]. *United States v. Microsoft* 253 F.3d (D.C. Cir 2001).

[34]. LIGUO ZHANG, “Refusal to license intellectual property rights under Art. 82 EC in light of standardisation context”, 32(8) E.I.P.R. 402-411(2010); THOMAS HEMPHILL & NICHOLAS VONORTAS, “U.S. Antittrust Policy, Interface Compatibility Standards, and Information Technology” 18 Knowledge, Technology & Policy 134 (2005).

[35]. Commentators including LEMELY, *supra* 31, recognise the benefits of interoperability while others consider the position is more ambiguous; *see e.g.* MARIO GIL-MOTO, “Economic aspects of the *Microsoft* case: networks, interoperability and competition”, in: LUCA RUBINI (ed.), “Microsoft on Trial” 344 at 359 *et seq.* (Edward Elgar, Cheltenham 2010).

[36]. Follow on innovation is dynamic rather than static competition *e.g.* coming within the description of dynamic competition advocated by GREGORY SIDAK & DAVID TEECE, “Dynamic competition in antitrust law”, 5(4) J.C.L. & E. 581-631 at 594 *et seq.* (2009).

[37]. URS GASSER & JOHN PALFREY, “When and how interoperability drives innovation” (31 October, 2007) http://cyber.law.harvard.edu/sites/cyber.law.harvard.edu/files/interopbreaking-barriers_1.pdf.

[38]. SMITA BRUNNERMEIER & SHEILA MARTIN, Research Triangle Institute, Center for Economics Research for NIST, “Interoperability Cost Analysis of the U.S. Automotive Supply Chain”, Final Report (March, 1999).

[39]. CARL SHAPIRO & HAL VARIAN, “Information Rules - a strategic guide to the network economy”, 107 (Harvard Business School Press, Harvard 1998).

40. DAVE DICKSON, “EPUB, iPAD and Content Interoperability Digital Editions”, http://blogs.adobe.com/digitaleditions/2010/01/epub_ipada nd_content_interope.html; (accessed 1 September 2010 on the Book Industry Study Group's, “Consumer Attitudes Toward E-book Reading”, (28 January 2010).

41. Section 1C(2) Registered Design Act 1949; Sec. 213(3)(b)(i) Copyright Design & Patents Act 1988.

42. **Software** Directive, *supra* note 1, Art. 1(2).

43. *Ibid.* at Art. 1(2) and recital 11.

44. *Ibid.* at recital 10.

45. *Ibid.* at recital 10.

46. *Ibid.* at recital 10.

47. WILLIAM R. CORNISH, “Interoperable systems and copyright, 11 E.I.P.R. 391-393 (1989).

48. American companies formed the **Software** Action Group for Europe (SAGE) and lobbied strenuously against Arts. 1(3) of the draft Directive; *see also* CAROLINE MEYER & MICHAEL COLOMBE, *supra* note 11.

49. *See* Arnold J in SAS Institute *supra* 3 at para. 205. The WCT entered into force with respect to the European Union and the United Kingdom on 14 March 2010.

50. Title 17 U.S.C. § 102(b) (1982).

51. *Whelan Associates Inc. v. Jaslow Dental Laboratory, Inc., et al.* 797 F.2d 1222 (3rd Cir. 1986).

52. *Computer Associates Int. Inc. v. Altai Inc.,* 982 F.2d 693 (2nd Cir. 1992).

53. Applies the levels of abstraction approach in *Nichols v. Universal Pictures Corp.,* 45 F.2d 119 (2nd Cir. 1930).

54. *Sega Enterprises Ltd. v. Accolade Inc.,* 977 F.2d (9th Cir. 1510). *See also Atari Games Corp v. Nintendo of America Inc.* 977 F.2d 1510 (9th Cir. 1992).

55. *Ibid.* at para. 1527.

56. Functional aspects were considered not to have copyright protection by Arnold J and AG Bolt in *SAS Institute, supra* notes 2 and 3, but the status of the **interfaces** was less clear.

57. *Lotus Development Corp. v. Borland International Inc.,* 49 F.3d 807 (1st Cir. 1995).

58. *Ibid,* at 818.

59. *Ibid,* at 816.

60. *Navitaire Inc v. easyJet Airline Co Ltd.,* (No. 3) EWHC 1725 (Ch) [2004]. The judgement in *Navitaire* was applied and approved by the Court of Appeal in *Nova v. Mazooma Games* [2007], EWHC Civ. 291.

61. *Ibid,* at para. 96.

62. *Ibid,* at para. 130.

63. Case C-393/09 *Bezpe#nostní softwarová asociace - Svaz softwarová ochrany v. Ministerstvo kultury* (2010), [2011] E.C.D.R. 3.

64. *Ibid,* at para. AG52.

65. *Ibid,* at para. 38.

66. *SAS Institute, supra* note 3.

67. *Ibid,* at para. 2.

68. Press release “World Programming Secures High Court Victory Against SAS David slays Goliath as 30 year monopoly is ended”. The High Court says that WPS is lawful clone of SAS system. http://www.teamwpc.co.uk/press/world_programming_secures_high_courtvictory_against_SAS (accessed 21 October 2011).

69. *SAS Institute, supra* note 3, at para. 69 where Arnold J found that WPS had never had access to the source code of the SAS System and there was no suggestion that WPL had decompiled any of the SAS object code or even attempted to.

70. *Ibid,* at para. 129.

71. *Ibid,* at para. 217 where Arnold J agrees with the ruling of Pumfrey in *Navitaire* that computer languages are not copyright protected although the expression of a program in a particular language may be protected. *See Navitaire, supra* note 60, at para. 88.

72. *Ibid,* at para. 129.

73. *Ibid,* at para. 205.

74. Proposal for a Council Directive on the legal protection of computer programs [1989] OJ C 91/4, at para. 3.12-3.13.

75. *SAS Institute, supra* note 3, at para. 226.

76. Proposal for a Council Directive on the legal protection of computer programs, *supra* note 3.13.

77. *SAS Institute, supra* note 3, at para. 248.

78. Opinion of AG in *SAS Institute, supra* note 2, at para. 50.

79. *Ibid.* at paras. 43 and 109.

80. *Ibid,* at para. 66.

81. *Ibid,* at para. 71.

82. *Ibid,* at para. 67.

83. *Ibid,* at para. 87.

84. *Microsoft, supra* note 4, at para. 287.

85. Joined Cases C-241/91 P and C-242/91 *Radion Telefis Eirann (RTE) and Independent Television Publications (ITP), Commission (Magill),* [1995] ECR I-743.

86. Case C-418/01 *IMS Health v. NDC Health* [2004] ECR I-5039.

87. *See* for example, ALAN DEVLIN, ET AL., *supra* note 22; *see also* SPULBER, *supra* note 22, at 249, describing the ways *Microsoft* adversely impacts incentives to innovate.

88. Microsoft argued that reverse-engineering provided the less advantageous yet “actual and potential alternative”. This standard of alternative to the input defeated the claim that the input was indispensable, *see* Case C-7/97 *Oscar Bronner,* [1998] ECR I-7791; and *see also* ARIANNA ANDREANGELI, “Interoperability as an “essential facility” in the Microsoft case - encouraging competition or stifling innovation?”, 34(4) E.L. Rev. 584-611, at 598 (2009), the notion of “indispensability” prevailing in *Microsoft,* emphasised the importance of interoperability to enable competing suppliers to operate viably on the relevant market, rather than focus on whether the protocols could be duplicated by competing suppliers.

89. The abuse in *Magill, IMS Health* and *Microsoft* all fall within the same category of abuse in Art. 105(b) TFEU “limiting production, markets or technical development to the prejudice of consumers”.

90. *Microsoft, supra* note 4, at para. 278.

91. *Ibid,* at para. 280.

92. *See also* SAMUELSON, *supra* note 10, at 1620, if reverse engineering is both lawful and feasible, trade secrecy protection for platform APIs is vulnerable.

93. *Microsoft* Commission Decision *supra* note 4, at para. Order 6.1.1.1 (999).

94. **Software** Directive, *supra* note 1, at recital No 10.

95. *Microsoft supra* note 4, at paras. 212 and 253.

96. Commission Decision in *Microsoft, supra* note 13, at 749-763.

97. *Microsoft supra* 4, para. 224.

98. *Ibid,* at para. 225.

99. *Ibid,* at paras. 227-228.

100. *Ibid,* at para. 265.

101. *See* GIFFORD & KURDRIE, *supra* note 24, at 708. Under US law the APIs in the Windows operating system are

"systems of operation" which are denied protection under Sec. 102(b) of the Copyright Act 17 A.S.C., so Microsoft sought to protect them as trade secrets.

102. Commission Decision in *Microsoft, supra* note 13, at para. 201.

103. ANN WALSH, *"Microsoft v. Commission:* interoperability, emerging standards and innovation in the **software** industry", in: RUBINI (ed), "Microsoft on Trial", *supra* note 35.

104. *Microsoft, supra* note 4, at para. 265.

105. Algorithms are excluded from copyright protection. While there are many ways to write code to perform a given task, the algorithms would vary less and protecting them would amount to protecting the function. This is more appropriate to patent protection. *See* Opinion of AG in *SAS Institute, supra* note 2, at para. 56.

106. IAN FORRESTER, *"Victa lacet mihi causa:* the compulsory licensing part of the *Microsoft* case", in: "Microsoft on Trial", *supra* note 35, at 97-98. The judgement meant Microsoft had to employ 210 **software** engineers and spend tens of thousands of hours to write the required specification.

107. Commission decision in *Microsoft, supra note* 13, at para. 244; *see also* COLIN JACKSON, "The basic technology issues at stake", in: "Microsoft on Trial", *supra* note 35, at 23. Kerberos had been developed by MIT and the specification for this protocol was in the public domain but Microsoft added a proprietary extension to the public standard.

108. Microsoft had 93% share of the PC operating system market and 60% of the work group server operating system market. Windows was described not only as dominant but as the "de facto standard". *See Microsoft, supra* note 4, at paras. 31-33.

109. *See Mirosoft, supra* note 4, at para. 1349 on interoperability leveraging quoting Bill Gates, "What we are trying to do is use our server control to do new protocols and lock out Sun and Oracle specifically. … Now, I don't know if we'll get to that or not, but that's what we are trying to do".

110. The Commission's position may have been informed by analysis in the *Microsoft* decision.

111. Case COMP/M.5669, *Cisco Systems, Inc. and Tandberg ASA.,* [2010] OJ L-2985.

112. *Ibid.* at para. 71. One important distributor (AT&T) said that Cisco, in acquiring Tandberg, would achieve better interoperability and if interoperability was made more difficult it would diminish the value of the new combined entity.

113. *Ibid,* at para. 81.

114. *Ibid,* at paras. 146-160. Cisco also undertook to make the source code of the TIP protocol available on an open source licence, ensure backward compatibility and support TIP for at least three years after the industry accepts a recognized standard.

115. Case COMP/M.5984 *Intel Corporation and McAfee, Inc.,* [2011] OJ L, at para. 18.

116. *Ibid,* at para. 160.

117. *Ibid,* at 62. The information seems to extend beyond interoperability to include information necessary to develop and optimise functionality in the Intel microprocessors and chipsets.

118. Press release Mergers: Commission clears Intel's proposed acquisition of McAfee subject to conditions, IP/11/70, 26/01/2011. *See also* GIFFORD & KUDRIE, *supra* note 24, where the appointment of Carl Shapiro's appointment as Deputy Attorney General in the Obama Administration is discussed and that subsequent action concerning Intel may indicate that the Federal Trade Commission sees no special antitrust status for the new economy.

119. *See* ZHANG *supra* note 34. *See also* WALSH, *supra* note 103 at 286.

120. WALSH, *supra* note 103, at 286.

121. The level of interoperability required from Intel and Cisco before their respective mergers were cleared was certainly more than could be achieved by reverse-engineering. *See e.g.* M.5984 *Intel/McAfee, supra* note 16, at para. 145.

122. WALSH, *supra* note 103, at 313, quoting J. FARRELL, "Standardization and Intellectual Property", 35 Jurametrics Journal 47 (1989).

123. **Software** Directive *supra* note 1, Art. 6(2)(b) although this restriction does not appear to apply to information gained by black box analysis under Art. 5.

124. WALSH, *supra* note 103, at 296-297.