



Legally Speaking

Do Software Copyrights Protect What Programs Do?

A case before the European Court of Justice has significant implications for innovation and competition in the software industry.

ARE THE BEHAVIORS of computer programs protectable by copyright law? What about programming languages and/or program interfaces? These questions were hotly debated in the U.S. from the mid-1980s to the mid-1990s. Several U.S. software companies argued that these aspects of computer programs were parts of the “structure, sequence, and organization” (SSO) of program texts that should be within the scope of protection that copyright law affords to programs. Courts in the U.S. ultimately rejected such claims. The European Court of Justice (ECJ) is faced with these questions in the pending *SAS Institute Inc. v. World Programming Ltd.* case.

SAS v. WPL

SAS is a well-known U.S.-based developer of an integrated set of statistical analysis programs, widely known as the SAS system. Base SAS, a core component of this system, enables users to write their own programs (known as scripts) in the SAS programming language. These scripts can then be executed on the SAS platform. SAS customers have written thousands of specialized programs, some of which are short and simple, others of which are large and complex, to perform specific types of statistical analysis tasks. Base SAS can be extended through use of other SAS components that support a broader range of functions.



The Grand Courtroom of the European Court of Justice.

World Programming Ltd. (WPL) recognized the possibility of a market demand for an alternative to the SAS system that could attract customers who had written scripts in the SAS language. The most important feature of an alternative platform would be its ability to allow these users to continue to run their scripts on the different platform. To test out this idea, WPL developed a set of statistical analysis programs designed to emulate the functionality of

the SAS system. Because the WPL programs accept the same inputs and produce the same outputs as SAS, scripts constructed in the SAS language can be executed by WPL’s software.

WPL had no access to SAS source code or to any proprietary design documentation for the SAS system. Nor did WPL decompile the SAS programs. It purchased copies of SAS software in the marketplace and studied manuals that came with SAS programs, tested

the SAS programs to study how they functioned under various conditions (that is, what inputs they accepted and what outputs they produced), and talked with SAS customers about their scripts and their needs.

After WPL launched its product, SAS sued WPL for copyright infringement for emulating the SAS program behavior, copying the SAS language, and reproducing SAS data formats. WPL argued that none of these aspects of the SAS system was within the scope of protection that copyright provides to computer programs. WPL won before the U.K. High Court, but the court referred the legal issues presented by the case to the ECJ for its interpretation.

Copyright in Program Behavior?

SAS's lawyer has argued that the detailed design of the functionality of the SAS programs is part of the SSO of the SAS programs, and as such, part of the expression that copyright law protects. Behavior is the intellectual creation of SAS, and a very substantial quantum of skill, labor, and judgment went into the design of the SAS program behaviors. A major goal of the European Directive on the Legal Protection of Computer Programs, the SAS lawyer noted, was to protect developers' investments in software. Allowing competitors to emulate a successful program's behavior would, he argued, undermine the first developer's opportunity to recoup these investments.

High Court Justice Arnold did not find SAS's argument persuasive. Although accepting that copyright protection extended to program source and object code, as well as to the architectural design of program internals and internal SSO, Justice Arnold stated that copyright did not extend to the functions that programs performed. The SAS program behavior consisted, in his view, of procedures, methods of operation, and mathematical concepts that U.K. law, consistent with European treaty obligations, deemed to be outside the scope of copyright protection.

In Justice Arnold's view, "[w]hat is protected by copyright in a literary work is the skill, judgment, and labor in devising the form of expression of the literary work," not the skill, judgment, and labor required to devise

ideas, procedures, methods of operation, and mathematical concepts that may be embodied in programs. This meant that WPL was free to copy the procedures, methods, and mathematical elements of the SAS programs.

The *WPL* decision relied quite heavily on an earlier U.K. High Court decision rendered by Justice Pumfrey, an experienced programmer, in the *Navitaire v. easyJet* case. Navitaire had developed an airline reservation program. easyJet had been one of Navitaire's licensees until it decided to commission the development of a "drop-in replacement" program that would accept the same commands and produce the same results as Navitaire's software.

As in the *SAS v. WPL* case, Navitaire sued easyJet for infringement and argued that program behavior was protectable by copyright law. Justice Pumfrey stated that a ruling in Navitaire's favor would be "an unjustifiable extension of copyright protection." Justice Arnold agreed with the reasoning in *Navitaire* that program functionality was not protectable by copyright law. Yet, he was sufficiently unsure of his interpretation of the European Software Directive that he referred this question to the ECJ.

Copyright in Programming Languages and Data Formats?

The programming language issue was difficult for both parties because of an ambiguity in the European Software Directive. It states that programming languages may be among the aspects of program-related innovations that should be considered ideas and principles that copyright law does not protect. But the Directive does not exclude

The programming language issue was difficult for both parties because of an ambiguity in the European Software Directive.

them from copyright protection altogether, but rather coyly indicates that *to the extent* that programming languages are ideas and principles, they are not protected by copyright law.

SAS's lawyer argued that the italicized phrase meant that programming languages were not automatically excluded from copyright protection, but could be protected against some types of appropriations. SAS's expert witness opined that the SAS language was not actually a programming language within the standard IEEE definition of that term, but rather a domain-specific language that was more like a command language, which SAS's lawyer argued would be protectable by copyright law.

WPL's expert witness testified that the SAS language was a programming language as that term is ordinarily understood in the computing field. He challenged as unsound the chart through which SAS's expert tried to distinguish among different types of programming languages.

Justice Arnold found the WPL expert's opinion to be more persuasive and more consistent with his understanding of the intent of the Software Directive. Although concluding that WPL's use of the SAS language was not an infringement of copyright either, Justice Arnold also referred interpretation of this aspect of the directive to the ECJ.

As for data formats, SAS's lawyer argued they were protectable expression as SSO embedded in the SAS programs. He argued that the SAS manuals studied by WPL had provided a window into this aspect of internal program structure, the copying of which should be considered infringement.

WPL argued the SAS data formats were interfaces whose use was necessary to the interoperability of scripts written in the SAS language with an independently written compatible platform such as WPL's. The Software Directive provides that elements of computer programs that are interfaces essential to interoperability are beyond the scope of copyright protection. WPL's reimplementation of these interfaces was thus non-infringing. Justice Arnold agreed with WPL on this point but referred the data format/interface issue to the ECJ as well. (SAS prevailed before the High Court on its claim that WPL manuals had

too closely paraphrased some wording from the SAS manuals. Changes to the wording of the manuals should be relatively easy to fix.)

Lotus v. Borland Redux?

SAS v. WPL raises the same legal issues as the *Lotus v. Borland* case, which was decided by a U.S. appellate court in 1995. Lotus was then the dominant seller of spreadsheet programs for use on PCs. Borland independently developed a spreadsheet program, which provided an emulation interface through which users could continue to execute macros for commonly executed sequences of functions that they had constructed in the Lotus macro command language. Use of the emulation interface enabled the Borland program to accept the same inputs as Lotus 1-2-3 and produce the same outputs, and hence, provided equivalent functionality or behavior as the Lotus program.

Borland's chief defense to Lotus claim that Borland infringed the Lotus copyright was that program behavior, macro language, and the command hierarchy that was an essential component of the Lotus macro system, were outside the scope of protection provided by copyright law. A U.S. appellate court ruled in Borland's favor.

One judge wrote a concurring opinion pointing to the lock-in effect for consumers of Lotus' position. Lotus wanted its users to be required to execute macros on its software. Borland provided those who had constructed macros in the Lotus language with an alternative platform on which to run the macros. As long as Borland independently created its program, it should not be held as an infringer of copyrights.

Since *Borland* and similar rulings, it has become well accepted in the U.S. that program "expression" does not include program functionality, but only what program texts say and how they say it. Interfaces are considered part of program functionality. Courts also recognize that reuse of command languages may be necessary for interoperability to occur, as in the *Borland* case.

Conclusion

In *SAS v. WPL*, an English High Court ruled that behavior is outside the

It has become well accepted in the U.S. that program "expression" does not include program functionality.

scope of protection that copyright law provides to computer programs. While the European Software Directive does not say anything directly about program functionality or behavior, it speaks of the need to find substantial similarity in the *expression* of program ideas before infringement is found. In the *SAS* case, as in *Borland*, WPL had no access to SAS source code or any other documentation about the internal design of the program, which is where program expression is to be found. So there has been no copying of program expression.

The High Court also ruled that programming languages and interfaces were beyond the scope of SAS's copyrights. The European Software Directive is quite explicit that program interfaces necessary to interoperability and programming languages should be among the aspects of programs that may be unprotectable ideas and principles.

SAS has argued that the Software Directive only excludes abstract ideas from the scope of copyright, not concrete innovations such as program functionality, languages, and data formats; it also argues that investment in software development will be harmed by a ruling in WPL's favor. These assertions are incorrect.

Competition and ongoing innovation in the software industry will be deeply affected by the outcome of this case before the ECJ. Let's hope the European court reaches the right conclusions on these important questions. **□**

Pamela Samuelson (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley.

Copyright held by author.

Calendar of Events

March 20–22

Performance Metrics for Intelligence Systems, College Park, MD, Contact: Elena R. Messina, Email: Elena.messina@nist.gov

March 21–23

International Conference on Intercultural Collaboration 2012, Bengaluru, India, Contact: Ravi K. Vatrappu, Email: vatrappu@cbs.dk

March 23–24

Consortium for Computing Sciences in Colleges Southwestern, Stockton, CA, Contact: Michael Doherty, Email: mdoherty@pacific.edu

March 24–28

10th Annual IEEE/ACM International Symposium on Code Generation and Optimization, San Jose, CA, Contact: Carol T Eidt, Email: carol@caliphernia.com

March 25–30

Aspect-Oriented Software Development, Potsdam, Germany, Contact: Robert Hirschfeld, Email: robert.hirschfeld@gmx.net

March 26–29

2012 Spring Simulation Multiconference, Orlando, FL, Contact: El Aarag Hala, Email: helaarag@stetson.edu

March 27–28

6th Simulation Workshop, Redditch, United Kingdom, Contact: Simon JE Taylor, Email: simon.taylor@brunel.ac.uk

March 28–30

Eye Tracking Research and Applications, Santa Barbara, CA, Sponsored: SIGCHI and SIGGRAPH, Contact: Dr. Carlos Hitoshi Morimoto, Email: chmorimoto@gmail.com