An American Perspective on the SAS v. WPL Case

Kluwer Copyright Blog
July 12, 2011

Guest Blog by Pamela Samuelson, Berkeley Law School

Are programming languages, program functionality, and data interfaces protectable by copyright law or not? These questions were highly contentious in the United States during the mid-1980s to mid-1990s. Plaintiffs in several cases argued that because these were parts of the “structure, sequence, and organization” (SSO) of programs, they should be within the scope of copyright protection accorded to programs as “literary works.” The EU is now confronting these same questions in the SAS Institute v. World Programming Ltd. case which is pending before the European Court of Justice (ECJ).

U.S. appellate court decisions in the Lotus v. Borland and Computer Associates v. Altai cases answered these questions with a resounding “no.” Borland provided users of its spreadsheet program with an emulation interface through which users could execute macros that they had had created in the Lotus macro command language; this caused the Borland program to produce the same functionality or behavior as the Lotus program. Altai had extracted interface information from Computer Associate’s code and reimplemented the interface in independently written code that operated in a substantially similar way.

The U.S. courts ruled that these aspects of programs were outside the scope of protection that copyright law affords to software developers. Since the Borland and Altai rulings, it has become well-accepted that program “expression” does not include program functionality, but only what program texts say and how they say it. Interfacing 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.

Lotus and Computer Associates argued that unless courts construed copyright law to give a broad scope of protection to program SSO, investment in software development would decline. The U.S. software industry has, however, thrived since the Borland and Altai rulings. The Business Software Alliance, a U.S.-based software trade association, has reported that the U.S. software industry contributed $260 billion to the U.S. economy in 2007. So the investment-based argument for broad protection is weak.

These same controversies have now surfaced in the EU in the SAS case. The English High Court decision is consistent with the Borland and Altai decisions. It found that WPL had not infringed copyright in SAS’s statistical analysis program when it reimplemented SAS program functionality in an independently written program. It held that WPL’s program, which was not infringing, was functionally identical to the SAS program. The court also found that the SAS file formats functioned as interfaces necessary to successful interoperability with the SAS program. In short, WPL’s program accepted the same inputs as the SAS program and produced the same outputs and emulated SAS program functionality (although WPL’s program does not support some features of the SAS program).

The English High Court was, I believe, right that its ruling was consistent with the EU software directive. It found support for its conclusions in parts of the directive indicating that program interfaces necessary to interoperability and programming languages may be among the aspects of programs that should be considered unprotected ideas and principles.

While the directive does not explicitly state that program functionality or behavior is protectable, it speaks of the need to find substantial similarity in the expression of program ideas before infringement is found in the directive. The directive does not say anything directly about program functionality or behavior, but it does require that the program differences be found in the program’s own structure, sequence, and organization.

SAS has argued that the 1991 EU software directive only excludes abstract ideas, not concrete innovations such as program functionality and command languages. It cites a recent decision by a U.S. court that raised the same issue. But this decision is not on point here, for it is concerned with infringement of the concrete idea to produce substantially similar functionality. Competence and ongoing innovation in the software industry will be deeply affected by the outcome of this case before the ECJ.

Let’s hope the ECJ gets to the right answers.