THE PAST, PRESENT, & FUTURE OF SOFTWARE INTEROPERABILITY

Pamela Samuelson, Berkeley Law

ECIS 21st Anniversary Event

December 1, 2011

LOOKING BACK

- It was far from inevitable—though this may not be obvious to those of you who were not part of the struggle in late 1980s/early90s—that copyright protection would be unavailable for program interfaces (IFs)
- After all, they are typically complex and detailed, requiring skill, judgment, & creativity to develop, seemingly satisfying copyright's originality standard
- Major US developers—notably IBM Corp.—were vigorously promoting copyright protection for IFs; ECIS was formed to explain why reimplementing IFs good
- Left to their own devices, copyright professionals might well have been receptive to IBM's claims
- But competition law & policy provided important counterweights, and ECIS helped to shape SW directive

STATE OF PLAY TODAY

- US & EU have seemingly adopted the same legal rules on interoperability, albeit via different routes
  - Interface information necessary to achieving interoperability is not within the scope of protection that © law provides to programs
  - If reverse engineering is necessary to get access to interface information, it is lawful to do this
- Lest we be complacent, however, there are possible perturbations to watch out for
  - *SAS v. WPL* before ECJ, *Oracle v. Google* in US

HOW DID WE GET HERE?

- Some US developments predated EU developments and had some influence on the SW directive, in part because of fierce lobbying by some US firms on software IP issues insisting EU must be consistent with US developments
- EU competition authorities’ experiences with IBM’s abuse of dominant position through changes to interfaces that rendered previously compatible products incompatible helped to inform SW directive too, as did ECIS arguments
- Brief review of this history may help put the *SAS v. WPL* dispute in context

CONTU (1979)

- At the time the CONTU Commission was deliberating about © & new technologies, the int’l conversation on SW protection mainly focused on a sui generis approach
- CONTU asserted that © was already available to software, but to allay any ?s, it recommended some amendments to © to clarify this + allow backups, bug-fixes, etc., enacted in 1980
- CONTU did not anticipate interoperability issues
  - Nor did it speak about language or behavior
  - But sec. 102(b) excluded processes, methods of operation, & systems from © which became impt

APPLE v. FRANKLIN (1983)

- Franklin developed “clone” of the Apple II computer, copied Apple II-OS, bit for bit
- Franklin argued that it was necessary to copy OS exactly to be compatible with programs written for Apple II
  - Hence, program ideas & expressions had “merged”
- CT AP: “Franklin may wish to achieve total compatibility with [Apple II applications], but that is a commercial & competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas & expressions have merged.”
- But Franklin did not even try to develop own programs to implement Apple II-OS functionality

WHELAN v. JASLOW (1986)

- Jaslow hired Whelan to help him automate common dental lab business processes
  - He was the domain expert, had taught himself to code
  - But she was a more expert programmer
- After a falling out, Jaslow developed a competing program
  - J used different programming language, different algorithms, designed SW to run on different computer than W’s
  - But overall structure was similar, as were data and file formats, & 5 subroutines performed similarly (“look & feel” a/k/a program behavior)

WHELAN’S ARGUMENTS

- SW = literary work
  - SSO of novels & plays is protectable by ©
  - SSO of software should also be protectable
- Without protection for program SSO, too little incentive to invest in program development
- 102(b) = restatement of idea/expression distinction under which only high level abstractions outside ©
- Similarities were too detailed to be “ideas”
- Other ways to structure the programs, so infringement to make substantially similar choices
- Not a case about interoperability but concept of “SSO” was so broad, it seemingly extended to IFs & behavior
  - SAS’s arguments in *WPL* case are quite similar

PROBLEMS WITH *WHELAN*

- Did not distinguish between high and low level detail SSO, or between different types of SSO, such as methods of operation excluded by 102(b)
- Did not recognize that sometimes structural similarities in programs may be due to their implementation of the same functional process, also excluded by 102(b)
- Ignored competition & innovation policy consequences of very broad SSO ruling
- Along with anti-compatibility dicta from *Franklin*, it was not looking good for interoperability in US in late 1980s
- *Whelan* was promoted by IBM et al. in EU debates as supporting IF protection by ©

EU DEVELOPMENTS (1988-89)

- 1988 EC “green paper” considered both © & sui generis forms of protection for software
  - Recommended © as primary vehicle
- Commission proposed directive on legal protection of computer programs in 1989
  - Initially did not exclude program interfaces from scope
  - No reverse engineering exception either
- SAGE for IBM & some other US firms argued that IFs were © SSO, also argued reverse eng’g was illegal in US
  - USTR Hills (IBM board of directors): must be © for IFs, no RE
- ECIS argued for reverse eng’g, vs. IF protection to allow independent development of interoperable programs

1991 SOFTWARE DIRECTIVE

- SW protected as literary works, including preparatory designs; protected vs. reproductions & adaptations
- Recital defined IFs as parts of programs that provide for interconnection of SW &/or HW
- Art. 1(2) states that ideas & principles that underlie programs, including those that underlie IFs, not protectable by © law
- Art. 6 recognized that IF info is sometimes not available, might be necessary to decompile a program to extract IF info; OK to do this in order to develop non-infringing SW
  - Art. 9(1) ensures this privilege cannot be overridden by K
- Reasonable interpretation: interfaces necessary for achieving interoperability not within © scope

SOFTWARE DIRECTIVE

- Art. 5(3): right of lawful user to study or test functioning of program to determine ideas & principles (implicit: right to reuse these)
  - This too cannot be overridden by K under Art. 9(1)
- Recital 14: to the extent logic, algorithms, & programming languages comprise ideas & principles, they are not protectable aspects of programs
- Directive does not say anything about program functionality or behavior, but focus on “literary” aspects of software suggests not
- Remarkably little litigation about SW directive
  - *SAS v. WPL* likely to be important precedent

POST-SW DIRECTIVE DEVELOPMENTS IN US

- Computer Associates v. Altai (1992) was first definitive ruling on IF issue in US
  - CA relied heavily on *Whelan*, arguing that interfaces were SSO protected by ©
  - But CT AP ruled that IFs are unprotectable
- Sega Enterprises v. Accolade (1992) was first appellate court ruling on decompilation for purposes of achieving interoperability
  - Trial judge ruled decompilation was infringement
  - But CT AP ruled it was fair use when done for legitimate purposes such as achieving interop'ity
- Had these decisions been rendered before SW directive adopted, directive might have been stronger on IF exclusions & legitimate reasons to reverse eng'r SW

ALTAI

- List of services & parameter similarities were to be expected because both CA's and A's programs needed to interoperate with IBM OS's
  - This was a constraint on programmer choices by external factors
  - Other similarities were as to abstract or public domain
- CA's arguments for broad protection were inconsistent with *Feist v. Rural Publications*, with *Baker v. Selden* & progeny, & with 102(b)
- CT AP: maybe functional design elements of program SSO should be patented

SEGA v. ACCOLADE (1992)

- Accolade reverse-engineered Sega code to discern how its interfaces worked so it could sell games for the Genesis console
- Sega argued RE copying was infringement
- Accolade won on fair use:
  - Purpose: to extract functional requirements for achieving interoperability = unprotectable elements of program, relying on *Altai* re no © for IFs
  - Nature of work: SW does not reveal unprotectable elements so RE necessary
  - Amount: copy whole thing, but necessary to extract
  - Harm: not commercially exploiting RE copies
- Accolade copied one segment of code that was essential to interoperability; court ruled this did not infringe

9th Cir in *SEGA*

- “If disassembly of copyrighted object code is per se an unfair use, the owner of the copyright gains a de facto monopoly over the functional aspects of his work—aspects that were expressly denied copyright protection by Congress. In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator of the work must satisfy the more stringent standards imposed by the patent laws.”

LOTUS v. BORLAND (1995)

- Same legal issues as *SAS v. WPL*
  - Borland’s Quattro Pro emulated the functionality of Lotus 1-2-3
  - Borland copied the Lotus macro language from L’s manuals, reimplemented in independently written code the Lotus macro system so that Lotus customers who had constructed macros could continue to execute them if switched to QP
  - Copying of command hierarchy was necessary to enable macro compatibility
- Relying on *Whelan*, DCT ruled vs. Borland
- CT AP reversed, rejecting *Whelan*-like argument
  - Borland only copied un-©’ble methods of operation

SAS v. WPL

- UK High Court ruled that WPL did not infringe SAS © because similarities were due to unprotectable elements:
  - Program behavior/functionality
  - SAS programming language
  - Interfaces
- WPL offered consumers an independently written program that could accept same data inputs & produce same outputs
  - Scripts written by SAS customers could be executed on WPL's alternative platform
- Art 5(3) allows black-box reverse eng'g; 9(1) K restriction null & void
  - SAS's effort to override this by contract was unavailing

ARNOLD ON SW DIRECTIVE

- © protects SW as “literary work”
  - Functional behavior of SW is not part of program “expression,” not analogous to plots of novels
  - SW directive should be construed in light of TRIPS which excludes methods of operation & mathematical concepts in SW
  - SAS behavior = methods of operation
- SW Directive identifies programming languages as among the elements that may be unprotectable ideas
  - Satisfied that SAS language qualified as programming language that should be excluded from © in this case
- Legislative history of SW directive reveals that IFs were intended to be excluded from ©
  - Data formats & syntax were functioning as IFs in this case

AG BOT’S OPINION

- Program functionalities are not part of program’s expression under the SW Directive:
  - “To accept that a functionality of a computer program can be protected as such would amount to making it possible to monopolize ideas to the detriment of tech’l progress & industrial development.”
- Interprets SW Directive as excluding programming languages from the scope of ©:
  - “A programming language devises specific methods to be used and facilitates the thinking necessary in order to write & formalize computer source programs.”
  - Programming language is “the means which permits expression to be given, not the expression itself”

AG BOT ON INTERFACES

- Notes that data formats are akin to blank forms to be filled in by customers
- Somewhat muddled about the SW Directive as to IFs
  - Directive does not exclude interfaces from © protection as such
  - Yet it indicates that ideas & principles underlying IFs may be unprotectable
- Erroneously assumes that WPL decompiled SAS code to get access to IF information
  - High Court expressly found that no decompilation had occurred (p. 69)
- Yet, it indicates (p. 90) that it may be OK to reproduce file formats if necessary to interop'ity and expression from the program is not copied

TYING ISSUES TOGETHER?

- In order for scripts written in SAS language to interoperate with SAS platform or alternative platform (e.g., WPL’s), data inputs must be in the precise format designed by SAS to yield same outputs
  - Thus, they are serving as program IFs
- WPL has emulated SAS program behavior to enable those who have written scripts in the SAS language to have these scripts interoperate
  - The language is thus an essential element to the interoperation of the scripts with the platform
  - No copying of expression from the SAS program

ANALOGY TO *EC v. MS*?

- *SAS v WPL* similar to the one-way v. two-way interoperability issue in EC case vs. Microsoft
  - MS was arguing that SW directive only meant to allow one-way interoperability (e.g., writing an application program that could interoperate with an OS)
  - EC contended that SW directive was meant to allow two-way interoperability (ie, Sun’s OS could emulate the functionality of MS’s OS and allow programs written for the MS OS to run on the Sun OS as well)
- *SAS* case is similar
  - SAS wants a rule that limits SAS scripts to interop’g with its platform
  - WPL wants to interoperate with SAS scripts & be drop-in replacement for SAS platform

CONCLUSION

- *SAS v. WPL* will be an important milestone in EU SW © law
  - Arnold, J. articulated not only reasonable interpretations of SW Directive, but also a persuasive application of these principles to the case at hand
- Although SW Directive does not flat-out exclude IFs and prog'g languages from ©, it should be understood to direct courts to pay close attention to whether the pro-interop'ity principles of the SW Directive call for withholding or extending © protection to certain aspects of programs which parties claim are IFs or languages
  - Goal overall was to allow the independent development of interoperable programs, good for competition & ongoing innovation in EU as well as in US
- Future of interoperability is bright, thanks to ECIS