UC Berkeley UC Berkeley Previously Published Works

Title Copyrightability of Java APIs revisited

Permalink https://escholarship.org/uc/item/2hs8t4z4

Journal Communications of the ACM, 58(3)

ISSN 0001-0782

Author Samuelson, Pamela

Publication Date 2015-02-23

DOI 10.1145/2723669

Peer reviewed

Legally Speaking: Copyrightability of Java APIs Revisited

Pamela Samuelson

For more than twenty years, the prevailing view has been that application program interfaces (APIs) are unprotectable elements of copyrighted computer programs. Under this view, programmers are free to reimplement other firms' APIs in independently written code. Competition and innovation in the software industry has thrived amazingly well in part because of rulings upholding this understanding.

Challenging this view is the Court of Appeals of the Federal Circuit (CAFC) May 2014 decision in *Oracle v. Google*. The CAFC held that the "structure, sequence, and organization" (SSO) of the Java APIs that Google reimplemented in its Android software are protectable expression under copyright law. It reversed a lower court ruling that the Java APIs were not copyrightable.

Google has asked the U.S. Supreme Court to review the CAFC's ruling. Several amicus curiae (friend of the court) briefs have been filed in support of this effort. Hewlett-Packard, Red Hat, and Yahoo! are among these amici (as am I and 77 computer scientists).

The Supreme Court may take the case because the CAFC's decision is in conflict with other appellate court rulings that exclude APIs from copyright protection.

This column will explain the Oracle and Google theories about the copyrightability of Java APIs and the precedents on which each relies. The stakes in this case could not be higher.

ORACLE'S CLAIMS

Developing Java APIs required considerable creativity. Sun's engineers had substantial freedom in the choices they made about how to structure the APIs. The Java APIs are thus easily original enough to qualify for copyright protection, says Oracle (which acquired the intellectual property (IP) rights in Java when it acquired Sun Microsystems).

Java has achieved considerable success, which is why Google wanted to use Java APIs in its software platform for mobile devices. Google entered into negotiations with Sun about licensing rights in Java, which shows that it knew that it needed a license.

When these negotiations failed, Google went ahead and copied 37 of the Java APIs anyway in the Android platform for mobile devices. Tens of thousands of Java programmers have written apps to run on that platform. These apps have contributed to the extraordinary success of Android devices.

Shortly after acquiring Sun and its assets, Oracle sued Google for copyright infringement. (There were originally some patent claims in the case as well, but a jury ruled against those claims.) Oracle relied on some judicial precedents that had held that the SSO of programs is protectable by copyright law as long as there are multiple ways to design that SSO.

SECTION 102(b)

At issue in the *Oracle* case is the proper interpretation of Section 102(b) of U.S. copyright law. It states that "[i]n no case does copyright protection for an original work of authorship extend to

any idea, procedure, process, system, method of operation, concept, principle or discovery, regardless of the form in which it is . . . embodied in such work."

Oracle asserts that this provision restates the classic distinction between expression (which copyright law protects) and ideas (which are beyond the scope of copyright protection). Because the Java APIs are much more detailed than ideas and may have original elements, they are not ideas alone, but rather expressions of ideas. The CAFC agreed, concluding that these Java APIs are copyrightable because of the creativity they embody and the existence of alternative ways in which Google could have developed its own APIs.

NINTH CIRCUIT PRECEDENTS

Google has pointed out that the plain language of Section 102(b) makes procedures, systems, and methods of operation unprotectable by copyright law. It asserts the Java APIs at issue are unprotectable under this provision.

Google has relied on several appellate court decisions to support its claims that the Java APIs are unprotectable by copyright law. Especially relevant is the Ninth Circuit Court of Appeals' ruling in *Sega v. Accolade*.

Sega sued Accolade because it made copies of Sega software in the course of reverseengineering to get access to the interface procedures embedded in the Sega code. Accolade needed to know this information to make its videogames compatible with the Sega platform.

The Ninth Circuit held this reverse engineering was a noninfringing fair use because it was done for the legitimate purpose of getting access to interface procedures which were "the functional requirements for [achieving] compatibility" and consequently unprotectable under Section 102(b).

Google claims that the CAFC erred by ignoring this aspect of the *Sega* decision. (Ordinarily an appeal from a California federal court would have gone to the Ninth Circuit, but because Oracle originally sued Google for patent as well as copyright infringement, Oracle's appeal from the copyright loss went to the CAFC instead. The CAFC was supposed to follow Ninth Circuit precedents.)

The CAFC opined that Google's arguments about compatibility might be relevant to its fair use defense to Oracle's claim of infringement, but not to whether the Java APIs were protectable by copyright law.

ORIGINS OF SECTION 102(b) EXCLUSIONS

Copyright's exclusion of systems and methods of operation from the scope of its protection traces back to the Supreme Court's 1880 ruling in *Baker v. Selden*. Selden sued Baker because he copied the bookkeeping forms Selden published to illustrate how to implement his new bookkeeping system. Selden won at the trial court level, and Baker appealed.

The Supreme Court perceived the question in *Baker* to be "whether the exclusive property in a system of book-keeping can be claimed, under the law of copyright, by means of a book in which that system is explained[.]"

The Court ruled that Selden's copyright extended to his *explanation* of the bookkeeping system, but not to the *system* itself, the method of operation it prescribed, or the forms that implemented the system. Such a "useful art" might have been eligible for patent protection, but not for copyright.

The Court observed that "[t]o give to the author of the book an exclusive property in the [useful] art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright."

Congress codified the *Baker* holding in Section 102(b). A legislative report said it did so "to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law."

The Ninth Circuit in *Sega* recognized that copyright law should not protect interface procedures because that would confer patent-like protection on the functional requirements for compatibility without Sega meeting the stricter standards required for patents.

The trial judge in *Oracle* expressed concern that Oracle's copyright claim might be seeking to obtain "an exclusive right to a functional system, process, or method of operation that belongs in the realm of patents, not copyrights." The court noted that "[b]oth Oracle and Sun have applied for and received patents that claim aspects of the Java API."

In overturning that decision, the CAFC, seemed untroubled about possible overlaps of copyright and patent protection for APIs. In effect, it read the procedure, system, and method exclusions out of the statute.

IS SSO PROTECTABLE BY COPYRIGHT?

The idea that program SSO is protectable expression as long as there is more than one way to accomplish a programming objective derives from a 1986 Third Circuit ruling in *Whelan Associates v. Jaslow Dental Lab.* Oracle and the CAFC have embraced this theory.

The SSO concept was, however, substantially discredited in the Second Circuit's 1992 *Computer Associates v. Altai* decision. In the years since *Altai*, courts have largely moved away from conceiving of SSO as protectable expression in programs because it fails to provide a workable framework within which to distinguish protectable and unprotectable structural aspects of programs.

The Second Circuit in *Altai* emphasized that the "essentially utilitarian nature of computer programs" makes it difficult to separate protectable and unprotectable structural elements in programs.

Altai announced a new "abstraction, filtration, and comparison" test for software copyright infringement. Among the structural elements of programs that must be filtered out before assessing infringement are efficient design elements, elements constrained by external factors, and standard programming techniques.

The Second Circuit in *Altai* was quite explicit that elements of programs "dictated by external factors" such as "compatibility requirements of other programs with which a program is designed to operate in conjunction" lie outside the scope of protection that copyright provides to programs. Such structural similarities must be filtered out before courts can determine whether a defendant infringed copyright.

The Ninth Circuit followed *Altai*'s lead in holding that interface procedures necessary for achieving interoperability among programs were functional elements of programs that copyright did not protect under Section 102(b). In *Sega*, the court cited approvingly to *Altai* for the proposition that computer programs "contain many logical, structural, and visual display elements that are dictated by the function to be performed, by considerations of efficiency, or by external factors such as compatibility requirements and industry demands."

LOTUS v. BORLAND

Another important appellate ruling that supports Google's theory is *Lotus v. Borland*. In 1995, the First Circuit ruled that Borland had not infringed by copying the SSO of the Lotus 1-2-3 command hierarchy for use in the emulation interface of Borland's Quattro Pro program. Borland had to use the same commands in the same order so that users who had constructed macros of frequently executed functions in the Lotus macro language could continue to use those macros in the Borland program.

The First Circuit in *Borland* did not find the SSO concept helpful in distinguishing protectable and unprotectable structural elements of computer programs. It held that the SSO at issue in *Borland* was an unprotectable method of operation under Section 102(b), akin to the command structure of VCR machines.

The trial judge in the *Oracle* case relied on the *Borland* decision, characterizing the Java APIs as a similar type of command structure. The CAFC chose not to follow *Borland*, and interpreted *Altai* as applicable only when initial designers of APIs are themselves constrained in their choices about structuring the interfaces.

CONCLUSION

Twenty years ago, the Supreme Court took Lotus' appeal from the First Circuit ruling. After oral argument, it split 4-4 on the proper interpretation of Section 102(b) as applied to computer programs. This left the First Circuit opinion intact, but did not make a nationwide precedent. The issues left undecided in that case are before the Court in the *Oracle* case.

Several amicus briefs filed in support of Google's appeal say that if the Supreme Court does not repudiate the CAFC's interpretation of copyright law, the result will likely be a new surge in litigation over the protectability of APIs, even though this issue had seemed to be resolved by appellate court rulings going back to 1992.

Oracle and its amici (if any) will have their say on the issues in briefs filed in December 2014. The Court will likely decide whether to hear the case in January or February of 2015. Because of

the importance of the issue to the software industry, I predict the Court will decide to hear the case. How it will rule remains to be seen.

About the Author:

Pamela Samuelson is the Richard M. Sherman Distinguished Professor of Law and Information at UC Berkeley. She can be reached at pam@law.berkeley.edu.