



DOI:10.1145/3466607

Pamela Samuelson

Legally Speaking

Reimplementing Software Interfaces Is Fair Use

Considerations for copyright protection to program interfaces.

A LONG-STANDING, GENERALLY ACCEPTED NORM in the computing field distinguishes between software interfaces and implementations: Programmers should have to write their own implementing code, but they should be free to reimplement other developers' program interfaces. This norm, of which Sun Microsystems, the developer of Java, was once the software industry's foremost proponent, is now the law of the land in the U.S. after the Supreme Court's decision in *Google Inc. v. Oracle America, Inc.*, overturned a lower court ruling that reimplementing an interface infringed copyright.

The Supreme Court took Google's appeal on two issues. One was whether program interfaces are protectable by copyright law. The Supreme Court declined to decide that issue, even though many amicus curiae (friend of the court) briefs filed by software developers, organizations such as the Electronic Frontier Foundation, the Center for Democracy & Technology, and the Computer & Communications Industry Association, as well as numerous in-

tellectual property scholars, supported Google's argument that program interfaces are uncopyrightable.

A second issue was whether Google's reimplementation of 11,500 declarations from 37 Java Application Program Interface (API) packages in its Android smartphone platform was fair use or infringement. Although a jury rendered a verdict in favor of Google's fair use defense after a two-week trial, the Court of Appeals for the Federal Circuit (CAFC) overturned this verdict. The CAFC concluded that no reasonable jury could have found Google's appropriation of that many lines of computer code was fair use. This is the ruling that Supreme Court's decision reversed.

The penultimate sentence in Justice Breyer's opinion for the 6-2 majority succinctly states the Court's conclusion: "where Google reimplemented a user interface, taking only what was needed to allow users [that is, programmers] to put their accrued talents to work in a new and transformative program, Google's copying of the Sun Java API was a fair use of that material as a matter of law."

After explaining Oracle's claims against Google, this column reviews the Court's reasons for rejecting Oracle's arguments on the fair use issue.

Oracle's Claims

Oracle's main argument on the copyrightability issue was that Sun's engineers had exercised considerable creativity in articulating, naming, and organizing the Java API declarations in an overall taxonomy of Java API packages, classes, and methods. As creative elements of the Java Standard Edition, Oracle contended they were eligible for copyright protection. It characterized the declarations as "declaring code," saying these lines of code were equally as copyrightable as implementing code.

Oracle's arguments against Google's fair use defense were more multifaceted. Google's purpose in copying the Java API declarations, in its view, was commercial and non-transformative (that is, Google used the declarations for exactly the same purpose as if Oracle or Sun had licensed this use). Both considerations, Oracle asserted, tipped against fair use.

Because Google went ahead and used the Java API in Android after its negotiations with Sun for a license to use Java technologies fell through, Google had acted in bad faith. Furthermore, Google had taken unfair advantage of the popularity of those declarations with experienced Java programmers. Applications that Java programmers have developed for the Android platform violate the “write once, run everywhere” objective of Java.

Also tipping against fair use, said Oracle, was Google’s exact copying of 11,500 lines of highly creative declaring code, which was quantitatively and qualitatively substantial. This taking had caused Oracle to suffer market harm because Oracle should have been getting substantial license fees from Google for its use of the Java API.

The phenomenal success of the Android platform had made it impossible for Oracle to enter or to license Java for smartphone platforms. Oracle claimed it was entitled to \$9 billion in damages due to Google’s misappropriation of the declarations.

The CAFC found Oracle’s arguments persuasive on virtually every point.

The Jury Verdict Mattered

One reason the Supreme Court overturned the CAFC’s fair use ruling was because the CAFC substituted its judgment for the jury’s about Google’s fair use defense. Appellate courts are supposed to defer to jury verdicts unless they are demonstrably wrong. The Supreme Court decided that the CAFC’s anti-fair use judgment was demonstrably wrong, not the jury’s.

By taking an extremely narrow view of the jury’s role in fair use cases, the CAFC had, in effect, usurped the legitimate role of the jury. Justice Breyer’s opinion took that court to task for ignoring evidence that supported many of Google’s arguments, including evidence showing that Google’s use of the Java API had not caused Oracle to suffer cognizable market harm.

Differences Between Declaring and Implementing Code

In most fair use cases, the nature of the copyrighted work factor is quite insignificant. In the *Google* case, the Court elevated its significance by starting its fair use analysis with a discussion of this factor.

The Court characterized the Java API declarations as a “user interface” because it was a way for programmers to control performance of specific tasks by computers through a set of commands. The Court characterized Java API declaring code as having different kinds of capabilities and as “embody[ing] a different type of creativity” from implementing code.

Declaring code was, moreover, “inextricably bound up with the use of specific commands known to programmers as method calls” that Oracle does not claim as copyrightable. Yet declarations are also “inextricably bound with implementing code, which is copyrightable, but which was not copied.” The Court viewed implementing code as much more conventionally expressive in a copyright sense than declaring code.

Also significant was that the value of declaring code derived chiefly from the investments that Java programmers had made in learning them so they could create new programs.

The Court also noted that Sun’s business strategy had been to make APIs open and to compete on implementations. Oracle may have adopted

a different strategy, but that didn't change Sun's history of promoting open interfaces.

Google's Purpose

The Court viewed Google's purpose in using the Java declarations in a positive light. It characterized the Android platform as "a highly creative and innovative tool for a smartphone environment." This platform also enabled experienced Java programmers to take advantage of their investment in learning the Java declarations to invoke common tasks when developing new programs. The Court regarded this as the very kind of creativity that copyright law was designed to promote.

The jury had, moreover, heard evidence that reimplementations of APIs was common and accepted in the software industry as long as developers reimplemented the interfaces in independently written code. Sun executives had believed the widespread use of Java would benefit it.

Substantiality of the Taking?

Although 11,500 lines of code may seem like a lot, the Court pointed out the Java API consists of 2.86 million lines of code. Google had used only 0.4% of the total amount. Those lines of code were, moreover, a small part of the 15 million lines of Android code.

Moreover, Google had produced evidence it copied only the declarations that Java programmers needed to write code for the Android smartphone platform. This new platform was different in kind than the laptop and desktop platforms for which the Java API had initially been developed.

Also relevant was that Google had copied the 11,500 declarations "not because of their creativity, their beauty, or even (in a sense) because of their purpose." Rather, it did so "because programmers had already learned to work with the Sun Java API system." It would be difficult "to attract programmers to build its Android smartphone system without them." Use of the Java declaring code "was the key [Google] needed to unlock the programmers' creative energies."

Market Harm?

Oracle's market harm arguments were unavailing, in part because the "jury

Although 11,500 lines of code may seem like a lot, the Court pointed out the Java API consists of 2.86 million lines of code.

could have found that Android did not harm the actual or potential markets" for the Java interfaces.

The failed negotiations with Sun had been over much more than the use of the 37 packages Google used in Android. Hence, Oracle's lost license fee argument failed.

The Court pointed to trial testimony that Sun's efforts to enter the mobile phone market with Java had been failures. Google's economic expert had explained that Android was "part of a distinct (and more advanced) market than Java software." Because the jury found Google's use of the Java declaration to be fair use, it must have decided against Oracle's market harm claims.

Significantly, the Court concluded that to enforce Oracle's claim of copyright infringement against Google would "risk harm to the public." In particular, it would make reimplementations of APIs "a lock that limited the future creativity of new programs," thereby "interfer[ing] with, not further[ing], copyright's basic creative objectives." Hence, the Court concluded Google's use of the Java declarations was fair use as a matter of law.

What About Copyrightability?

The controversy over whether program APIs can be copyrighted dates back to the 1980s. When the issue was finally litigated in the late 1980s and early 1990s, several appellate courts ruled that reimplementations of interfaces, when necessary to achieving compatibility with other programs, was not copyright infringement. The CAFC's ruling on the copyrightability of APIs in *Oracle v. Google* was contrary to more than 20 years of other court precedents on this issue.

Google, along with dozens of amicus curiae briefs filed in support of Google's

legal positions, would have preferred for the Court to have decided the *Oracle* case on copyrightability grounds. In an amicus curiae brief, 71 intellectual property scholar colleagues and I argued that APIs should be deemed unprotectable methods or procedures as a matter of copyright law, as pro-interoperability lower appellate court decisions had done.

A common reason why software industry briefs urged the Court to rule on the copyrightability issue was that this ruling would provide greater certainty in the software industry so that developers could reimplement others' interfaces free from fear of lawsuits.


Fair use, by contrast, is typically very fact-intensive and decided on a case-by-case basis. Developers do not want to have to spend precious resources fighting litigation over their reimplementations of APIs.

Conclusion

The Court's *Google v. Oracle* analysis of the fair use issue is strongly favorable to fair use defenses involving reimplementations of APIs and hints in several places that APIs may not be copyrightable.

The Court's decision leaves untouched several appellate court rulings that denied copyright protection to program interfaces, even citing to some of them approvingly. Those decisions all involved situations in which true program-to-program interoperability was at stake.

Google v. Oracle was not as convincing as a compatibility case because apps developed for the Android platform do not necessarily run on other Java platforms. And apps written for Java-compliant laptops do not necessarily work on Android. Fair use was, consequently, a better defense than challenging their copyrightability.

In view of the Court's decision and the intact precedents in true interoperability cases, I predict there will be very few software interface infringement cases brought any time soon. Software developers can now breathe a deep sigh of relief as Oracle's aggressive claims have been debunked. 

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

Copyright held by author.