



Legally Speaking

Oracle v. Google: Are APIs Copyrightable?

Assessing the first phase of the trial based on claims that Google's Android platform infringes Oracle's Java-related copyrights and patents.

ARE APPLICATION PROGRAM interfaces (APIs) of computer programs protectable by copyrights in software that embodies them? *Oracle v. Google* is the most definitive ruling yet that addresses this question because the judge took pains to understand exactly what Java APIs are, how and why Google implemented them, and how the copyright dispute between these two software giants meshed with software copyright precedents.

The judge rejected Oracle's claim of copyright in Java APIs and his ruling suggests that APIs are uncopyrightable more generally. Oracle will appeal, but Judge William Alsup of the U.S. District Court of Northern California did a very careful job in analyzing the issues. I predict affirmance.

Origins of the Lawsuit

In 2005, Google began making plans to create a platform for mobile phones. It held some discussions with Sun Microsystems about a possible license to use, adapt, and open source Java for mobile devices. These negotiations proved inconclusive, so Google went ahead with its Android project without Sun's involvement. Google decided to use the Java language to design its own virtual machine and to write its own implementations of 37 of the 166 Java APIs.

The Android platform was released in 2007. Android-based mobile phones went on the market the next year. More than 300 million of these phones have



Courtroom sketch depicting Google attorney Robert Van Nest questioning Oracle CEO Larry Ellison during *Oracle v. Google*.

been activated since then. Google provides the Android platform for free to smartphone manufacturers and other developers. It makes money on smartphone ads and search.

Although Sun knew that Google was using Java without a license, it did not sue. Shortly after Oracle acquired Sun in 2010, Oracle sued Google for copyright infringement in a California federal court. (There were patent claims as well, which the jury rejected.)

Oracle's main copyright claim was that Google's Android platform infringed copyright because it copied the "structure, sequence, and organization" (SSO) of 37 Java APIs without permission. As the new owner of Java assets, Oracle had the right to sue anyone who it thought infringed its rights.

In August 2011, Google tried to avoid going to trial about its use of Java APIs by moving to dismiss that part of Oracle's lawsuit. If APIs are not pro-

tectable by copyright law, as Google believed, a trial was unnecessary.

By denying Google's motion, the judge seemed to give some credence to Oracle's claim. He seemed to give it further credence when he sent the API claim to the jury, asking it to decide two things: first, whether, *assuming that APIs were copyrightable*, had Google infringed by copying them, and second, whether this copying was fair use. The jury answered the first question in the affirmative and split 9-3 on the fair use question.

In the week or so after the judge dismissed the jury, everyone interested in the *Oracle v. Google* case was on tenterhooks. Because he was taking so long to rule on the legal question, the judge gave the impression that Oracle's claim might have some merit.

Yet in view of his eventual ruling, it now appears the judge was just being careful. The trial gave him a chance to understand what APIs were, what uses Google had made of the Java APIs, and why these uses were important to interoperability. Once he understood this, Oracle's claim was doomed.

Oracle's Arguments

Here is why Oracle thought its API copyright claim might succeed. In addition to making much ado about a Google engineer's belief that Google needed a license to implement Java, Oracle relied upon some appellate court decisions upholding SSO infringements.

In *Whelan v. Jaslow*, for instance, a small software company sued the owner of a dental lab because the latter wrote a program for managing dental lab operations similar to that which he had commissioned from this firm. Jaslow's program was written in a different programming language and used different algorithms. But the overall structure of his program was substantially similar to Whelan's and five subroutines were substantially similar in operations.

Because copyright law considers computer programs to be "literary works" and because copyright protects the SSO of conventional literary works, the *Whelan* court reasoned that the SSO of programs should be protectable also. The court thought that if copyright was only protected against exact copying of another's

code, as Jaslow claimed, this would undermine incentives to invest in software development.

A key issue in *Whelan* concerned the proper interpretation of section 102(b) of U.S. copyright law: "In 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 described, explained, illustrated, or embodied in such work."

Jaslow argued that he had only copied the unprotectable methods and procedures of Whelan's program, not any expressive aspects of that software. Whelan argued that because Jaslow had copied detailed structures from her program, he had copied expression. Section 102(b), in Whelan's view, merely restated the rule that copyright does not protect abstract ideas, but only expressions of ideas. The appellate court agreed with Whelan's interpretation. Several other cases relied on *Whelan* in affirming infringement rulings based on program SSO similarities.

Another factor that seemed to support API copyrights was that developing APIs requires some creativity. Oracle asserted that Java APIs were creative enough to meet copyright's originality standard. APIs are also valuable aspects of programs and software companies have been known to recoup at least some of the costs of program development by licensing interface specifications and SSO to other companies. Moreover, if one took section 102(b) literally, computer programs would not be copyrightable, for they are by their very nature functional processes. Yet Congress clearly intended that programs should be protected.

Google had ample ammunition with which to counter Oracle's legal arguments.

Oracle analogized Java API names and name groupings to taxonomy names and groupings that had been held copyright-protectable in some non-software cases. Oracle pointed out that it was unnecessary for Google to use the same function names as Java or to arrange the names in the same way.

Because the Android platform was not fully interoperable with Java, Oracle challenged Google's compatibility defense. Moreover, by implementing only 37 of the 166 Java APIs, Oracle complained, Google was contributing to fragmentation of Java and undermining the "write once, run everywhere" goal of Java.

Google's Arguments

Google had ample ammunition with which to counter Oracle's legal arguments. The *Whelan* case, on which Oracle so heavily relied, has been discredited by a later much-cited appellate court decision, *Computer Associates v. Altai*. *Altai* criticized *Whelan* for having an outmoded conception of computer programs and as relying too heavily on metaphysical distinctions rather than practical considerations. Because *Whelan* did not involve copying of APIs, its precedential value was weak on that score as well.

Altai ruled that if a subsequent programmer's design choices were constrained by external factors, such as the hardware or software with which the program was designed to interoperate, copyright law would not penalize similarities attributable to this. Although the *Altai* court did not specifically mention APIs, it observed that parameter list similarities between Computer Associates's and *Altai*'s scheduling programs were noninfringing because they were due to constraints imposed by the IBM operating system programs with which both litigants' programs were designed to interoperate.

Also supporting Google's API defense were two other appellate court decisions, *Sega v. Accolade* and *Sony v. Connectix*. Both ruled that reverse engineering another firm's software for a legitimate purpose such as seeking access to information necessary for interoperability was non-infringing.

Accolade was free to use information obtained by reverse engineering to adapt its video games to run on

Sega's Genesis platform. The court in *Sega* characterized interface information as "functional requirements for achieving compatibility with other programs," which was unprotectable under section 102(b). Connectix could lawfully develop an alternative software platform on which consumers could play video games designed for the Sony PlayStation.

The Court's Analysis in *Oracle v. Google*

Although none of these cases specifically mentioned APIs, their import for the API copyrightability question in the *Oracle* case was clear.

The judge was also influenced by the *Lotus v. Borland* decision, which rejected Lotus' argument that its choice of command names and the arrangement of those commands in a hierarchy was "expressive." The court held that Lotus' menu command hierarchy, which was a fundamental part of the functionality of the Lotus macro system, was an unprotectable method of operation within the meaning of section 102(b).

Based on these cases and his understanding of Java APIs implemented in Android, the judge concluded that the Java APIs were unprotectable methods: "[A]nyone is free under the Copyright Act to write his or her own code to carry out exactly the same function or specification of any methods used in the Java API. It does not matter that the declaration or method header lines are identical. Under the rules of Java, they must be identical to declare a method specifying the same functionality—even when the implementation is different."

As for the Java names and name groupings, the judge observed that "the names are more than just names—they are symbols in a command structure...Each command calls into action a pre-assigned function." The Java command structure, the judge went on to say, was an unprotectable system under section 102(b), duplication of which "is necessary for interoperability." Hence, Google had not infringed copyright by its use of the Java APIs.

Oracle's concerns about fragmentation of Java were, he ruled, irrelevant to the copyright issues in the case.

When one company buys another, it is not bound by the legal positions previously taken by the acquired company.

Conclusion

There is irony in Oracle's API copyright claim against Google. When Sun was a major player in the computer industry, it was among the most vigorous proponents of pro-interoperability intellectual property rules. Sun, for instance, submitted amicus curiae briefs in the *Altai* and *Lotus* cases taking exactly the opposite position from Oracle in the *Google* case. Sun also supported an exclusion of software interfaces from copyright protection in Europe.

When one company buys another, it is not bound by the legal positions previously taken by the acquired company. Yet it should be more judicious in making contrary claims than Oracle was. Former Sun employees and lawyers may be among those smiling at Oracle's having been hoisted by its own petard. The *Oracle* API copyright ruling is a big victory for Google, but an even bigger one for competition and ongoing innovation in the software industry.

The month of May this year was a good one for interoperability. In addition to the Oracle API ruling, the Court of Justice of the EU rendered its decision in the *SAS Institute v. World Programming Language case* (about which I wrote in my March 2012 *Communications* column). It ruled that copyright protection for software does not extend to the functional behavior of programs, to programming languages, or to interfaces necessary for interoperability.

As a result, it is now safer than ever to develop interoperable programs on both sides of the Atlantic Ocean. 

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

November 22–23

Annual Workshop on Network and Systems Support for Games, Venice, Italy,
Contact: Maha Abdallah,
Email: maha.abdallah@lip6.fr

November 26–30

The 24th Australian Computer-Human Interaction Conference, Melbourne, Australia,
Contact: Vivienne Farrell,
Email: vfarrell@swin.edu.au

November 28–29

Conference on Visual Media Production, London, U.K.,
Contact: Oliver Grau,
Email: oliver.grau@bbc.co.uk

December 2–6

HILT 2012: High Integrity Language Technology: ACM SIGAda Annual, Boston, MA
Sponsored: SIGAda,
Contact: Benjamin M. Brosgol,
Phone: 646-375-0730,
Email: brosgol@gnat.com

December 3–7

13th International Middleware Conference, Montreal, Canada,
Contact: Bettina Kemme,
Phone: 514-398-8930,
Email: kemme@cs.mcgill.ca

December 10–13

Conference on Emerging Networking Experiments and Technologies, Nice, France
Sponsored: SIGCOMM,
Contact: Renata Cruz Teixeira,
Email: renata.teixeira@lip6.fr

December 18–20

Third Kuwait Conference on e-Services and e-Systems, Kuwait,
Contact: Saleh Kassern,
Email: kasserns@yahoo.com

January 6–8

ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA,
Contact: David S. Johnson,
Phone: 908-582-4742,
Email: dsj@research.att.com