

COPYRIGHT COMMENT

Google v Oracle: important questions



Hillel I Parness

Is copyrightability of computer code headed back to the Supreme Court of the US? **Hillel I Parness** examines what might happen if or when it gets there, and what in-house counsel should do in the meantime

On 12 January 2015, the Supreme Court of the US (SCOTUS) invited the US Solicitor General to file a brief in connection with Google's petition for certiorari in *Google, Inc v Oracle America, Inc.*¹ This article discusses what is at issue, what Google and Oracle have told the Supreme Court, and what may happen. The article concludes with a set of practical questions that all companies should be asking right now, to ensure that they and their programmers do not run afoul of the Copyright Act, regardless of the outcome of Google's petition.

Factual and procedural overview

The *Google v Oracle* case concerns undisputed copying by Google of portions of Sun's (now Oracle's) source code in Google's creation of the Android operating system:

"Sun wrote a number of ready-to-use Java programs to perform common computer functions and organised those programs into groups it called 'packages'. These packages, which are the application programming interfaces at issue in this appeal, allow programmers to use the prewritten code to build certain functions into their own programs, rather than write their own code to perform those functions from scratch. They are shortcuts."²

As described by the Federal Circuit, Oracle denied Google's request for a licence to the application programming interface (APIs), because Google refused to make its programs interoperable with other non-Google Java programs. Google instead designed its own Java-based system but still used the 37 APIs. Thus, as the Federal Circuit put it, "it is undisputed that Google copied 7,000 lines of

declaring code and generally replicated the overall structure, sequence, and organisation of Oracle's 37 Java API packages", and the "central question before us is whether these elements of the Java platform are entitled to copyright protection".

At the trial court, the jury found that Google had infringed Oracle's copyrights in the 37 APIs, but in a subsequent decision, the District Court for the Northern District of California found that the APIs were not copyrightable, and thus, could not be infringed.

Federal Circuit decision

The Federal Circuit reversed on this issue, holding that "the declaring code and the structure, sequence, and organisation of the API packages are entitled to copyright protection... with instructions to reinstate the jury's infringement finding as to the 37 Java packages," and also remanding on the question of whether Google enjoys a fair use defence.

The Federal Circuit made a number of key rulings on its way to finding the 37 APIs copyrightable, including:

- **Merger:** The court began by addressing the merger doctrine – that when idea and expression merge, the expression is unprotected. Merger, wrote the Federal Circuit, is a defence to infringement in the 9th Circuit, and therefore irrelevant to copyrightability. Furthermore, because there had been multiple ways for Oracle's programmers to write the 37 APIs, the court said that the merger doctrine did not apply. The Federal Circuit also pointed out that the "multiple ways" question is determined at the time the original work was created, not at the time of infringement, so if Google felt constrained to use the APIs, it is irrelevant.
- **Scenes a faire:** The court likewise rejected

the application of the scenes a faire doctrine – the non-protectability of stock phrases – both because it is a defence to infringement, and because Google had not shown that at the time of programming, the programmers employed stock phrases of code that everyone needed.

- **Lotus:** The Federal Circuit found the Supreme Court's 1996 affirmance without opinion in *Lotus v Borland*³ did not stand in the way of the copyrightability of the 37 APIs, because computer code that embodies creative expression, where the programmer had choices in how to write the code, can be protected under copyright. "Google, like any author", wrote the court, "is not permitted to employ the precise phrasing or precise structure chosen by Oracle to flesh out the substance of its packages – the details and arrangement of the prose."
- **Interoperability:** The court also rejected Google's argument that the copying was necessary to achieve interoperability, returning again to the question of whether the programmers had choices at the time of programming: "[t]he compatibility Google sought to foster was not with Oracle's Java platform or with the JVM central to that platform. Instead, Google wanted to capitalise on the fact that software developers were already trained and experienced in using the Java API packages at issue." The court further rejected Google's argument that the APIs could be copied because they were "the effective industry standard", pointing out that "Google cites no authority for its suggestion that copyrighted works lose protection when they become popular, and we have found none."
- **Fair use:** Consistent with the Federal Circuit's approach, it remanded the question of fair use to the trial court, because it had

Hillel I Parness is the founder of Parness Law Firm, where he focuses on complex commercial litigation and intellectual property. He also serves as a member of the adjunct faculty at Columbia Law School, where he teaches courses related to intellectual property and the internet.

not been decided previously. Google's interoperability, the court held, was irrelevant to the question of copyrightability, but it could factor into a fair use analysis.

The parties' briefs on certiorari

Google has petitioned SCOTUS to grant certiorari, even before the fair use question returns to the trial court, on the following question:

Whether copyright protection extends to all elements of an original work of computer software, including a system or method of operation, that an author could have written in more than one way.

Google leads with arguments based on the *Lotus* case, characterising it as reflecting an unresolved split among the Circuit courts, as to whether "methods of operation embodied in computer programs" are or are not protectable under copyright law. Google analogises the QWERTY keyboard layout to the Java conventions embodied in the APIs:

"People invested time and effort in learning the QWERTY design, and then expected all keyboards to use it... Programmers have made significant investments in learning these commands; they are, in effect, the basic vocabulary words of the Java language. When programmers sit down to write applications, they expect to be able to use them."

Google argues that the need for compatibility, and the concern of avoiding "lock-in", means that computer code like the APIs cannot be restricted under copyright law:

"If one must use specific computer code in order to operate computer programs such as the pre-written programs at issue here, that means, almost by definition, that the copied code is part of a system or method of operating the programs." Throughout its briefs Google reiterates the concern that protecting common methods will stifle innovation, at one point analogising the Java APIs to the standard elements of all cars, and referring to both as "crucial methods for operating a complex system".

Oracle met Google's various analogies with one of its own:

"What Google did was the equivalent of plagiarising the topic sentence of every paragraph of a blockbuster novel, as well as the chapter and subchapter titles, and then paraphrasing the rest.

Google's argument – that the code lost all copyright protection because it became popular and Google wanted Android to be popular, too – would not fly for any other work. It does not fly for computer code."

Oracle argues that "no circuit disagrees that 7,000 lines of original and creative code are copyrightable", and explains how the *Lotus* case and others consistently hold that expression, not raw ideas, can be protected. Oracle also states that it does not seek to use copyright law to prohibit others from writing their own code that achieves the same functionality, noting that "[a]nyone is free to write software that performs the same functions and embodies the same ideas as Oracle's."

"SCOTUS is unlikely to say that computer code is not entitled to protection, but rather to expound upon the idea/expression dichotomy through the facts of this case."

Oracle also responds to Google's argument on interoperability by repeating that "Google designed Android so that it would not be compatible with the Java platform."

Predictions

Conventional wisdom suggests that SCOTUS' invitation to the Solicitor General to weigh in indicates that the court is interested in the Solicitor General's view, and further suggests that when the Solicitor General does comment in such situations, the court often acts in a manner consistent with the Solicitor General's recommendations. In the *Cartoon Network v CSC Holdings* case regarding remote DVRs, for example, the Solicitor General recommended that SCOTUS deny the petition for certiorari, and that is what the court did. Until we see whether and how the Solicitor General responds to the invitation, all of this is mere speculation.

As for how the Supreme Court would rule in an ultimate appeal, requires a different type of speculation. The court's prior decisions in *Metro-Goldwyn-Meyer Studios, Inc v Grokster, Ltd* (1995) and *American Broadcasting Companies, in et al v Aereo, Inc* (2014), both suggest a pragmatic approach to high-stakes copyright cases, based in large part on the defendants' motives and their use of copyright precedent to advance the same. If the case goes up the court, we may see both Google and Oracle advance similar pragmatic arguments – Google may say that Oracle may not use copyright law to prevent Google from building upon the advances of prior programmers, while Oracle may argue that Google may not avoid the basic application of copyright law and the licensing scheme by labeling the Oracle code as necessary for interoperability. In the author's view, SCOTUS is unlikely to say that computer code is not entitled to protection, but rather to expound upon the idea/expression dichotomy through the facts of this case.

Practice points

Regardless of what happens next, companies that create complex computer code are strongly advised, as always, to put procedures in place to avoid problems. In-house counsel and executives must know the answers to all of the following questions, and keep their information current:

- Who are the employees in the company that actually create or obtain code?
- What training have they received regarding computer code and intellectual property? What type of continuing training do they receive?
- Are your programmers using any types of preexisting programs or modules, like the Google programmers used the 37 Oracle APIs? If so, what procedures and infrastructure are in place for legal to become aware and review the relevant contractual provisions?
- The previous question and suggestions are equally applicable to issues of open source software.
- How do your programmers document their programming process? How would legal, in an audit, be able to determine where code came from?

Footnotes

1. Available at http://www.supremecourt.gov/orders/courtorders/011215zor_3e47.pdf.
2. *Oracle America, Inc v Google, Inc*, 2013-1021, -1022 (Fed Cir 9 May 2014).
3. 49 F3d 807 (1st Cir 1995), *aff'd without opinion*, 516 US 233 (1996).