



COMPLEXITIES EVIDENT IN TODAY'S ORAL ARGUMENT IN ORACLE V. GOOGLE

by JONATHAN BAND on DECEMBER 4, 2013

Today, the U.S. Court of Appeals for the Federal Circuit heard oral argument in Oracle v. Google. The complexities of the case were demonstrated by the fact that the oral argument lasted an hour and a half – thirty minutes longer than the extended time the court had already allocated for the argument.

In 2010, Oracle sued Google in federal district court in San Francisco, alleging that the Android Application Program Interface (API) infringed Oracle's copyrights and patents relating to the Java API. Oracle owned the intellectual property in Java by virtue of its acquisition of Sun Microsystems. In 2012, a jury found that Google had not infringed Oracle's patents. With respect to the principal copyright claim, Judge Alsup ruled that the elements of the Java API copied by Google – the method headers (a line or two of code at the beginning of each subroutine) and the organization of 37 Java API packages – were not protected by copyright under 17 U.S.C. § 102(b).

Oracle appealed to the Federal Circuit, asserting that Judge Alsup had erred by holding that the copied API elements were unprotectable methods of operation. Oracle further argued that Judge Alsup incorrectly found that program elements necessary for interoperability were not protected by copyright. Finally, Oracle asserted that Google had copied elements of the Java APIs not for the purpose of achieving interoperability, but rather to attract the community of Java developers to the Android platform.

CCIA filed an amicus brief in support of Google, arguing that over the past 25 years, U.S. courts, Congress, and jurisdictions around the world have arrived at a consensus that copyright protection does not extend to program elements, including interface specifications, necessary for achieving interoperability. In the judicial and legislative battles that resulted in this consensus, both Sun Microsystem and Oracle consistently had supported the pro-interoperability position, until Oracle reversed itself in this case.

The complexity of the case was obvious from the get-go, when Judge Taranto (who represented Grokster before the Supreme Court in MGM v. Grokster) asked Oracle's counsel, Joshua Rosenkranz, a series of questions about how an Android app developer would access the Android API. Judge Plager on several occasions expressed bewilderment on how a court should go about separating expression from functionality in a computer program, where every line of code is functional. Oracle's counsel and Google's counsel, Robert Van Nest, articulated a basic factual disagreement on whether an app written in Java could run on Android.

The panel wrestled with the question of whether the structure of the Java API was an unprotected method of operation. Of the three judges, Judge O'Malley appeared to lean most clearly towards Oracle's position that organization of the Java API was protected because it

reflected a high degree of creativity and Sun's programmers could have organized the API in a different manner when they designed. Google's lawyer responded that the Ninth Circuit (whose law will control because this case arose from a district court in the Ninth Circuit) in *Sega v. Accolade* had held that program elements necessary for interoperability were unprotectable. Judge O'Malley insisted that Sega only reached the issue of fair use, not protectability under Section 102(b). But on this point Judge O'Malley is incorrect. The Ninth Circuit stated that the district court found that Accolade reverse-engineered

"Sega's software solely in order to discover the functional requirements for compatibility with the Genesis console – aspects of Sega's programs that are not protected by copyright. 17 U.S.C. § 102(b)."

The Ninth Circuit's conclusion that Accolade's disassembly was a fair use was predicated on its holding that the information Accolade sought – "the functional requirements for Genesis compatibility" – was not protectable per Section 102(b). The Ninth Circuit explained that if reverse engineering were not permitted,

"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. 17 U.S.C. § 102(b). In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator must satisfy the more stringent standards imposed by the patent laws."

And while Judge O'Malley was correct that at the time Sun designed the structure of the Java API it could have designed it in a different way, she overlooked what the Java API is. It is not just a computer program or even a set of computer programs. It is an entire programming system Sun provided to programmers for the development of Java apps. And thousands, if not millions, of apps have been written using this system. Google replicated the structure of this system in Android, but not its implementation. Google copied the 7,000 lines of declaring code that embodied the "outline" of the Java API, to use Judge Taranto's phrase, but it wrote 15 million lines of original code that implemented that outline.

Even if one of the other judges agrees with Judge O'Malley that the Java API is protectable, that is not the end of the case. As all the judges acknowledged, Google's use of the Java API could be a fair use. In his concurring opinion in *Lotus v. Borland*, Judge Boudin posited fair use as an alternative to the rationale in Judge Stahl's majority opinion that the Lotus 1-2-3 command structure was an unprotectable method of operation. Oracle's counsel argued that the panel should decide that Google's actions were not a fair use. Google's counsel responded that the question of fair use should be remanded to the district court. The judges' questions suggested that they were leaning towards Google that the fair use issue should be remanded to the district court, assuming they find that the Java API structure was protectable.

Moreover, a ruling that the API structure was protectable could very well result in a visit to the Supreme Court, because such a ruling would conflict with the Ninth Circuit in *Sega* and the First Circuit in *Lotus*.

In short, this case could be far from over.

Jonathan Band is a DC-based attorney whose clients include Internet companies, providers of information technology, universities, library associations, and CCIA. He previously guest-posted on DisCo about misconceptions in copyright studies.

FEDERAL CIRCUIT RULES, ORDERS FURTHER TRIAL IN ORACLE V. GOOGLE

by MATT SCHRUERS *on* MAY 9, 2014

The Federal Circuit's Oracle v. Google decision over copyright in the Java APIs is out. If you're looking for a quick or simple resolution to this dispute, you'll be disappointed.

Back in December when the Oracle v. Google case was argued before the Federal Circuit, my colleague Jon Band (with whom I co-authored a brief in the case) offered some observations based on the oral argument, which ran a full 30 minutes late, as the 3-judge panel grilled the party's lawyers on questions of copyright, infringement, and fair use.

Jon noted the many complexities in the case, the continuing importance of the question of fair use, and predicted that the case was far from over. Today's opinion confirms all of those assessments: the Federal Circuit issued a lengthy, 69-page opinion, affirming in part, reversing in part, and also sending the case back to the trial court for an entirely new trial on the complex factual questions around the four fair use factors.

In short, the opinion reverses the district court's finding that the structure, sequence, and organization (SSO) of Java's APIs was not copyrightable, setting up an apparent circuit split that could go to the Supreme Court. It further holds that Google's use of Java API packages in Android infringed the Java copyrights, and also that Google's fair use defense needs further exploration.

The court's opinion begins its discussion by noting that it is "mindful that the application of copyright law in the computer context is often a difficult task", repeating the oft-cited quote from Judge Boudin in *Lotus v. Borland* that "Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit," but ultimately concludes that, in this case, the Java APIs were copyrightable, and infringed.

Stay tuned for further analysis from Jon...

THE FEDERAL CIRCUIT'S POORLY REASONED DECISION IN ORACLE V. GOOGLE

by JONATHAN BAND on MAY 9, 2014

As Matt reported earlier, the Federal Circuit issued a very disappointing decision today in the litigation between Google and Oracle that has the potential to upset well-established practices in the computer industry that encourage interoperability and competition.

The case is complex factually and procedurally. The main issue before the Federal Circuit was the protectability under copyright of elements of the Java Application Program Interface (API), which Google incorporated into the Android API. The lower court had ruled that the method headers (a line or two of code at the beginning of each subroutine) and the organization of the 37 Java API packages were not protected by copyright under 17 U.S.C. 102(b).

Following a lengthy oral argument last year, the Federal Circuit reversed, finding that these elements were protectable. The Federal Circuit remanded the case back to the trial court to determine whether Google's copying was permitted under the fair use doctrine (17 U.S.C. 107).

The opinion written by Judge O'Malley reflects a fundamental misunderstanding of the controlling precedent. On non-patent matters, the Federal Circuit must apply the law of the circuit from which the case arose. Here, because the case was decided by a court in the Ninth Circuit, the Federal Circuit is required to apply Ninth Circuit law.

The law in the Ninth Circuit could not be more clear: program elements necessary for interoperability are not protectable under copyright. In *Sega v. Accolade*, the Ninth Circuit found that the Accolade reverse engineered "Sega's software solely to discover the functional requirements for compatibility with the Genesis console – aspects of Sega's programs that are not protected by copyright. 17 U.S.C. 102(b)."

Nonetheless, because the *Sega* court issued this ruling in the context of determining whether Accolade's reverse engineering was a fair use, Judge O'Malley incorrectly held that the permissibility of copying interface information was a fair use question. She failed to appreciate that the Ninth Circuit's conclusion that Accolade's disassembly was a fair use was predicated on its holding that the interface information Accolade sought – "the functional requirements for Genesis compatibility" – was not protectable per Section 102(b). The Ninth Circuit explained in *Sega* that if reverse engineering were not permitted,

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. 17 U.S.C. §102(b). In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator must satisfy the more stringent standards imposed by the patent laws.

The Ninth Circuit's decision in *Sony v. Connectix* that fair use permitted the reverse engineering by Connectix likewise was predicated on the unprotectability of the interface information Connectix was seeking.

Additionally, Judge O'Malley brushed aside the First Circuit's ruling in *Lotus v. Borland*, which found that the Lotus 1-2-3 command structure was an unprotectable method of operation. Instead, Judge O'Malley relied upon the long discredited Third Circuit decision in *Apple v.*

Franklin that compatibility is “a commercial and competitive objective which does not enter into the ... issue of whether particular ideas and expressions have merged.” She further mischaracterized Borland as suggesting that computer programs could not receive copyright protection because they were functional works.

Judge O’Malley also completely distorted the policy arguments made by Google and its *amici*. She said that the arguments “appear premised on the belief that copyright is not the correct legal grounds upon which to protect the intellectual property rights of software programs.” Nothing could be further from the truth. Google and its *amici* made clear that they completely agreed that copyright protected software. The question before the court was whether copyright protected these specific program elements necessary for interoperability.

For the past 20 years, since decisions such as *Sega v. Accolade*, *Computer Associates v. Altai*, and *Lotus v. Borland*, computer programmers in the United States have understood that copyright does not protect program elements necessary for interoperability. Based on this understanding, programmers have freely copied these elements, which has encouraged enormous creativity, innovation, and competition in the digital environment. Judge O’Malley’s decision casts doubt on this understanding. By ruling that interoperability is relevant only to fair use, and not to protectability, Judge O’Malley would require every developer to perform a fair use analysis before developing an interoperable product. This would place U.S. programmers at a competitive disadvantage to developers in other jurisdictions that recognized that copyright does not protect program elements necessary for interoperability. The Court of Justice of the European Union, for example, in the 2012 decision in *SAS Institute v. World Programming Ltd.*, ruled that program functionality, programming languages, and data formats were not protectable under copyright.

One possible light at the end of the tunnel is that on remand, the lower court could rule that the copying of program elements necessary for interoperability is a fair use as a matter of law. After all, the Ninth Circuit in *Sega* held that “where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law.” Such a ruling would restore the legal certainty that Judge O’Malley has upended.

Update: Further reflections on this subject [follows].

Jonathan Band is a DC-based attorney whose clients include Internet companies, providers of information technology, universities, library associations, and CCIA. He previously guest-posted on DisCo about the Oracle v. Google oral argument.

FURTHER REFLECTIONS ON ORACLE V. GOOGLE

by JONATHAN BAND on MAY 12, 2014

This is the fourth post in a series on Oracle v. Google. [Previous posts (see above)]

I've now had the weekend to digest the Federal Circuit's decision in *Oracle v. Google*, and have some further reflections beyond those I expressed shortly after the decision issued this past Friday.

1. *The Federal Circuit's decision is not binding precedent in any other case in any district court anywhere in the country.* Because this case arose in the Ninth Circuit, the Federal Circuit was required to apply Ninth Circuit precedent. But its interpretation of Ninth Circuit precedent is not binding on district courts in the Ninth Circuit, or any other circuit. Thus, a district court in San Jose or Seattle is free to ignore the Federal Circuit's misinterpretation of *Sega v. Accolade* and *Sony v. Connectix*.

To be sure, other district courts may find the Federal Circuit's reasoning to be persuasive, but it is not binding. And as these district courts dig into the Federal Circuit's reasoning, they quickly will conclude that it is not that persuasive. The Federal Circuit certainly undermined its credibility with its assertion that Google and its amici believe that software should not be protectable under copyright. The Federal Circuit itself flatly contradicts this assertion when it observed that

Google agrees with the district court that the implementing code is the expression entitled to protection.... Indeed, at oral argument counsel for Google explained that, "it is not our position that none of Java is copyrightable. Obviously, Google spent two and a half years...to write from scratch all the implementing code."

The Federal Circuit noted that "of the 37 Java API packages at issue, '97 percent of the Android lines were new from Google.'"

2. *The Federal Circuit's statement that interoperability is irrelevant to determining protectability is dicta.* In my previous blog post, I explained that the Ninth Circuit had determined that elements necessary for interoperability fell on the idea (and therefore unprotectable) side of the idea/expression dichotomy, but that the Federal Circuit had misinterpreted these precedents. The Federal Circuit, however, went even further than stating that interoperability was irrelevant to the determination of whether an element was protectable; it also disagreed with the district court's finding that replication of the structure of the Java API was necessary to achieve a degree of interoperability between Java and Android. The district court found that in order for at least some of the existing Java apps to run on Android, Google had to provide the same command system using the same names with the same functional specifications. The Federal Circuit, in contrast, noted that there was "no evidence in the record that any such app exists." Moreover, the Federal Circuit claimed that Google could "point[] to no Java apps that either predated or post-dated Android that could run on the Android platform." The Federal Circuit further stated that Google did not seek to foster compatibility "with Oracle's Java platform or the [Java Virtual Machine] central to that platform."

If, in fact, Java and Android are not compatible, then the Federal Circuit had no reason to reach the issue of the impact of interoperability on the protectability analysis. Any statements on that question are mere dicta.

3. The court's explanation of what Java elements Google copied and why it copied them underscores that those elements are unprotectable under 17 U.S.C. 102(b), which withholds copyright protection from "any idea, procedure, process, system, method of operation, concept, principle, or discovery..." Section 102(b) is the codification of the long-standing idea/expression dichotomy that lies at the heart of copyright. The Second Circuit in *Computer Associates v. Altai* recognized that "drawing the line between idea and expression is a tricky business."

The Federal Circuit found that Google's real objective in replicating the structure of the Java API was

to capitalize on the fact that software developers were already trained and experienced in using the Java API packages at issue. The district court agreed, finding that, as to the 37 API packages, "Google believed that Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as in Java." Google's interest was in accelerating its development process by "leverag[ing] Java for its existing base of developers."

The Federal Circuit went on to say that

Google was free to develop its own API packages and to "lobby" programmers to adopt them. Instead, it chose to copy Oracle's declaring code and the [structure, sequence and organization] to capitalize on the preexisting community of programmers who were accustomed to using the Java API packages.

The Federal Circuit opined that the desire to capitalize on the preexisting community of Java programmers "has nothing to do with copyrightability." But this plainly is wrong. What could be better proof that something is a procedure, system, or method of operation than if a person can become "trained," "experienced," or "accustomed" to using it in the course of developing new works? Earlier in the opinion, the Federal Circuit described the API packages as "shortcuts" programmers can use when writing their own programs. While the detailed steps of each shortcut may be copyrightable, the structure of the entire set of shortcuts surely isn't. If a framework of shortcuts used by programmers in their development process isn't a procedure, system, or method of operation, what is?

Jonathan Band is a DC-based attorney whose clients include Internet companies, providers of information technology, universities, library associations, and CClA. He previously guest-posted on DisCo about his initial thoughts about the Oracle v. Google opinion.