Westlaw Journal
INTELLECTUAL PROPERTY
Litigation News and Analysis • Legislation • Regulation • Expert Commentary
VOLUME 19, ISSUE 7 / JULY 25, 2012

Expert Analysis

API and the Limits of Copyright Protection for Software

By Dr. D'vorah Graeser
Graeser Associates International

Software has always been a point of contention in the realm of copyright protection. When creating software, often information is collected from other works in order to create new software; because of this, the line between fair use and infringement has always been a narrow one.

As software programmers came to realize their creations were often artistic, elegant and highly profitable works, they turned to IP protection to support commercialization and to ensure credit for their hard work. Given the initial lack of certain types of IP protection, such as patents, software gurus turned to copyright, an imperfect solution, but as it presented streamlined approach to protecting what they deemed artistic works, they went with it.

And in truth, software can be protected by copyright in many different ways. For example, the code itself is protected by copyright. To a more limited extent, graphical user interface, or GUI, which determines the experience of the user when interacting with the software, is also protectable through copyright. Yet, since the first computer programs were sold commercially, the limits of copyright protection for software have been all too apparent.

Copyright protects the fruits of human creativity when provided in a fixed, tangible form. In fact, computer code was initially denied copyright protection as being either not creative or, alternatively, not tangible. Currently, software is copyrighted in the U.S. as literary works, clearly demonstrating that copyright law was not developed with software in mind. A recent case involving Oracle and Google, and their battle over the extent to which copyright protection enabled Oracle to dictate terms for use of its Java APIs (application programming interfaces), highlights the severe limits of current copyright law in terms of providing a desirable level of protection for computer code.

FIXED, TANGIBLE FORM

As previously noted, copyright law does not protect all forms of human creativity, but rather only protects works that have been fixed in a tangible medium of expression.

For example, choreographed dance steps must be written down to receive copyright protection (although they may also be protected through a recorded video performance of the work). However, dance was clearly originally intended to be experienced as a live performance; whether recording the performance or recording the steps in written form, copyright law effectively limits protection available to the dynamic, ephemeral original live performance.

Ideas, methods of operation, procedures and mathematical concepts are not copyrightable, although the first three may be protectable under patent law, if other requirements are met.

Software is defined and protected under U.S. copyright law. A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

This emphasis on the fixed tangible nature of the copyrighted work, as opposed to the experience of the consumer of the copyrighted work, is clearly to the detriment of certain types of creative works — among them software. The value in software clearly relates to the experience of the consumer of the software — either directly, in terms of the GUI or other interaction with the software, or indirectly, in terms of a desired result being provided to the consumer (for example, the operation of a mobile telephone).

Some of the earliest examples of this type of failure of copyright law relate to court cases involving the graphical user interface. In these cases, the GUI was separated into multiple different elements, which were then considered separately. While this may have made more sense for an analysis with regard to copyright law, it clearly is not the way in which a software user experiences the "look and feel" of the GUI.

This approach was most clearly made in *Apple Computer v. Microsoft Corp.,* 35 F.3d 1435 (9th Cir. 1994), in which Apple failed to defend its GUI copyright because the court held that those elements that were protectable under copyright and were used without authorization were so limited that only virtually identical expressions would be protected.

Thus, even though the consumer of software would be expected to have at least a similar experience when interacting with both the Apple and Microsoft GUIs, the fact that they aren't identical was enough to block copyright enforcement. (As an ironic note, Xerox sued Apple around this time, claiming that in fact Xerox originally developed the GUI; Xerox's claims were dismissed without the GUI copyright issue being considered.)

Other decisions, such as *Lotus Development Corp. v. Borland International,* 516 U.S. 233 (1996), held that elements of the GUI (in this case, the menu structure) were purely mechanical in nature, notwithstanding their effect on the user experience.

Although the Borland engineers testified as to their dedication to developing "better" software than existing accounting products on the market (including that of Lotus), they still copied the menu structure of the Lotus software. Indeed, Borland even offered a Lotus "emulation" GUI for its software. Apparently, the desire to present software consumers with familiar GUIs trumped any desire for improved software. The court, however, still held that the Lotus menu structure was not protectable by copyright, as it was mechanical in nature and not a creative expression.

API: THE GUI OF INTER-COMPUTER COMMUNICATIONS

So we come to the API. An API, application programming interface, enables two software programs to communicate with each other, thereby supporting inter-computer communication. Without access to an API for a particular software program, another program cannot communicate with it. APIs are the inter-computer equivalent of a GUI, as both permit interaction with a software program. APIs are clearly valuable to a software company, and control over these APIs is typically provided through licensing agreements with software developers.

For example, Garmin and eBay both have API licenses, which dictate when, where and how software developers may use their APIs. Garmin's license uses some terms that seem to indicate that it considers the APIs to be copyrightable (for example, by banning developers from creating "derivative works") but refers more generally to all "intellectual property" related to APIs.

Garmin maintains the right to terminate the developer's ability and right to use the APIs at any time, indicating that regardless of whether Garmin's rights are derived from copyright, Garmin feels that it has the right to control all uses of and interactions with the Garmin APIs.

Ebay goes even further, reserving the right to charge for "excessive" API calls, although without referring specifically to derivative works (the license does ban modifying or altering the APIs).

A recent court decision involving Oracle and Google specifically illustrates many of the inherent weaknesses of copyright protection for software. *Oracle Am. v. Google Inc.*, No. 10-3561, 2012 WL 1964523 (N.D. Cal. May 31, 2012).

Software companies including Oracle seem to feel that they have control over the APIs that they license, to the extent that they ban production of changed or derivative works based on such APIs. Since Oracle lost or withdrew the majority of its *patent* claims during its court battle against Google, such control can only be asserted through copyright — even if not explicitly stated by any licensing agreement.

In Oracle's case, the issue is further complicated by third-party APIs (not released by Oracle) that exist for Java. Google used some of this code to produce its own Java libraries, so that it did not copy the Oracle libraries.

Oracle argued that Google copied (a very few) lines of code and created a "non-standard" implementation of Java. In order for this non-standard implementation to operate with other programs written in Java, Google had to use the Java APIs — even if it avoided copying related Oracle code.

Google argued back that APIs are not copyrightable as they are purely mechanical in nature (sound familiar?) — more like a list of instructions than a creative work.

Oracle lost its argument with the court, although it can appeal the ruling. Using arguments and reasoning from previous GUI- related court cases, including the previously described Lotus case, the judge in this case held that the APIs are purely mechanical and are not copyrightable. Essentially, Oracle wanted the court to agree that the "look and feel" equivalent for APIs should be protected, while the court fell back on previous GUI-related arguments, failing to uphold such a broad interpretation of copyright.

The implications of the decision are broad as APIs are used more widely than ever to provide compatibility and communication between software programs and computers — of which mobile telephones are only the latest example. But the court reached back into years of software copyright decisions to yield a ruling in line with previous decisions. With Oracle's Java APIs still accessible, development of the feature-rich apps and programs being developed in the language will likely continue unabated.

Dr. D'vorah Graeser is the founder and CEO of **Graeser Associates International**, an international health care intellectual property firm. Graeser has been a U.S. patent agent for over 15 years and has extensive experience in the biomedical field. Her firm specializes in the preparation, filing and prosecution of medical device, biotechnology, pharmaceutical, bioinformatics and software patents. The firm also develops customized intellectual property strategies for companies with interest in selling their ideas and products internationally, primarily in the U.S. Europe, China, Brazil and India. For more information about Graeser and Associates International, please visit http://gai-ip.com/. Graeser can also be reached via LinkedIn or on Twitter @DGraeser.