

## Implementing Open Standards in Open Source

by Lawrence Rosen<sup>†</sup>

Industry standards morph into functional computer software. I use the word “morph” on purpose to avoid any term that can be found in US copyright or patent law. Morphing is a special effect in motion pictures and animation to turn one image into another through a seamless transition. Wikipedia shows an image of George W. Bush morphing into Arnold Schwarzenegger, and so too the morphing of an industry standard into software can result in something that looks entirely different at an expressive level and that potentially does useful things.

In the case of software industry standards, morphing transforms a written specification into working code through a mental process conducted internally by programmers and engineers. The end result – functional software – is a created outcome of human intellect that starts with a written specification and ends with a working implementation.

For attorneys, software specifications are unusual beasts. A specification may be the description of something patentable, but it is not itself patentable. Only an implementation of a specification, something that can be made, used, or sold, may be subject to patent infringement lawsuits (35 USC 271). Likewise, a specification itself can also be copyrighted, although the copyright does not 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 the specification (17 USC 102(b)). The rights to intellectual property in an industry specification (and thereby perhaps control over its intellectual content) are thus subject to some difficult legal questions with uncertain answers.

This topic gained added relevancy recently because of the patent and copyright infringement lawsuit by Oracle against Google relating to Java. We have been led to believe that Java is an industry standard for a programming language. In compliance with the rules of the Java Community Process (“JCP”), the Java community develops final specifications for technology to be included in the Java platform and publishes free implementations of those specifications under open source licenses. There are currently more than 300 published specifications for the Java language. There is much free and open source software implemented in compliance with these Java specifications. Many companies, individuals, and non-profit foundations (including the Apache Software Foundation of which I am a member) participate in the JCP with the goal that those specifications be available to all free of charge. The *Oracle v. Google* lawsuit has placed a patent and copyright cloud on Java specifications and software.

Specifications are different from software, but they are weapons in the competitive software wars and they are subject to legal control by contract and by law. Companies try to control specifications because they want to control software that implements those specifications. This is often incompatible with the freedom promised by open source principles that allow anyone to create and distribute copies and derivative works without restriction.

This article explores ways that are available to compromise that incompatibility and to make open standards work for open source.

### **Copyright on Industry Standards: Does Implementation Create a Derivative Work?**

The standard of proof for copyright infringement includes the element of “substantial similarity.” Because copyright protects expression, an infringing work must resemble the expression of the original, not merely its underlying ideas and concepts. If they do not look alike, it will be difficult to prove that an implementation is a derivative work of a specification. In the *SCO v. IBM* litigation, for example, the plaintiff was ultimately unable to identify any specific code in Linux that was derivative of its own UNIX software, even though they were functionally similar.

Often a specification and its implementation are very dissimilar. Specifications that contain English words describing functionality are intended to be implemented in C++ or Java or some other programming language, and the natural language description – certainly to the untrained eye – doesn’t resemble the resulting code in the slightest. The judges and juries that will determine copyright infringement won’t be able without expert advice to determine the substantial similarity of highly technical expressive works of software.

Some software experts view this dissimilarity of expression between specification and software code as a technical disadvantage, and so they are trying to create specifications that are their own software implementations. In the HTML5 project, for example, which is creating standards for structuring and presenting content in browsers on the World Wide Web, the drafters are experimenting with a specification technique that replaces specification text with public domain reference implementations in terms of an abstract state machine, in an attempt to improve compatibility by avoiding the imperfect conversion of English to source code. Implementers are encouraged to copy specification text as software. In these cases, at least, a software implementation is obviously a derivative work of the specification, and thus an infringement unless licensed.

Regardless of the style of specification writing and the programming language, it is to nobody's advantage to argue in court whether an implementation is a derivative work of the specification. A legal argument about whether an implementation is a derivative work, no matter how convincing, is not as convincing as a written license that expressly authorizes those derivative works. And so implementers seek licenses and standards organizations offer them. Standards bodies have devised their copyright licensing policies with the expectation that software implementations will be derivative works of their specifications.

Because software implementations will often be distributed under open source licenses, the compatibility of specification copyright licenses with open source rules becomes critical. The Open Web Foundation (OWF), in its specification license, addresses the copyright issue for software specifications succinctly and directly:

*I grant to you a perpetual (for the duration of the applicable copyright), worldwide, non-exclusive, no-charge, royalty-free, copyright license, without any obligation for accounting to me, to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, distribute, and implement the Specification to the full extent of my copyright interest in the Specification.*

Note that the copyright license here is "to the full extent of my copyright interest," thus effectively ignoring the question posed in the title of this section. Note also that the broad copyright grant to implement the Specification is fully compatible with all open source (and proprietary) licenses.

That copyright grant, however, is not satisfactory to some standards organizations. It goes too far with respect to allowing derivative works of a specification *as a specification*. These standards organizations seek to protect the purity of their specifications by forbidding other standards organizations to take over those specifications and modify them. In colloquial terms, and as they frequently justify this restriction, these organizations wish to prevent the "forking" of their specifications that might result in incompatible implementations.

This is a reasonable business and technical concern for a standards organization. Anyone who witnessed the early years of the browser software wars will remember how functionally incompatible browsers inhibited the development of advanced websites, and allowed commercial competition rather than technical benefits to dictate browser functionality. No software implementer wants to repeat that experience.

Compatibility requirements, however, are anathema to open source implementers. A requirement that all implementations function in a particular way

is contrary to every open source license that guarantees complete freedom to create derivative works. The desire of standards organizations to prevent forking of open standards contradicts the requirement of open source licenses that permit any derivative works.

This problem has recently been the subject of heated discussion in W3C relating to the new HTML5 specification. Nearly 80% of the W3C members responding to a survey said they do not want W3C to permit forking of W3C specifications, but they also overwhelmingly say that they want to encourage implementation of any open source software. By way of compromise, one proposal was for a new HTML5 license to allow software derivative works but to forbid specification derivative works. The Free Software Foundation, however, has argued that this restriction means that the proposed license is incompatible with the GPL. As this paper is being written, no final decision has been made about this license.

This problem was addressed by IETF in yet another way. They require specification writers to distinguish between "text" and "code." The IETF copyright license allows derivative works of the code but not the text portions of its specifications. Thus implementers may use the code portions, but if they seek to document those code portions they are not allowed to create derivative works of the IETF specification text. This rather arbitrary way of addressing copyright law issues of derivative works of specifications is based on distinctions between text and code that are not found in copyright law or in computer science. I do not believe this distinction is enforceable practically.

The *Oracle v. Google* lawsuit presents another aspect of this derivative work copyright problem. Oracle asserts in its complaint that Google has infringed Oracle's Java copyrights (presumably relating to the Java specifications, although the complaint in that case is not clear). Java specifications are published under a license that requires implementers to validate compliance with Java specifications using test compatibility kit (TCK) software licensed by commercial companies with contractual restrictions on the types of software derivative works that can be implemented. As such, it is incompatible with the requirements of the Apache License under which the Apache Software Foundation, and Google, publish their software. The Apache Software Foundation objected publicly to Sun, the Java specification steward, when it first imposed these contractual restrictions on the kinds of software derivative works that can be created if open source implementers license the Java TCK; now Oracle is the Java steward, and the concern has been reanimated by this recent litigation.

This question of whether software is a derivative work of a specification has thus become more important recently. Indeed, there may be no single answer that would satisfy those who create and seek to protect standards and those who

implement those standards under open source licenses. Agreement on specification licenses will require a degree of compromise over copyrights that neither commercial companies nor the open source community have yet achieved.

### **Patents on Industry Standards: Can a Specification Infringe a Patent?**

Copyrights are not the only intellectual property problem besetting software standards. Far more difficult are the effects of patents on the software ecosystem, because copyrights encumber only derivative works but patents can encumber *any* implementation. A clean room implementation of a specification will avoid copyright infringement -- but it is not so easy to avoid patent infringement.

Patents are not a problem for specification writers, but rather for implementers. Anyone is free to write a description of how to perform a process or method – indeed the patent system requires the open publication of just such a specification when a patent is granted. But when that specification is transformed into functioning software and distributed for actual use by actual people and companies, those users may be patent infringers. In the proprietary software world such risks are commonly borne by the vendors of software products with offers of limited warranties and indemnity, usually capped at reasonable dollar limits. But with free and open source software that is distributed without warranties or indemnities, the risk of patent infringement when using implementations of standard specifications is borne by the user.

Note that this patent risk is in practice quite low. While hypothetical situations can be litigated easily in the mind, there have been no notable patent infringement lawsuits in U.S. federal court over software standards implemented in open source software. When major software companies cooperate openly to develop industry standards in organizations like W3C and IETF, few of them have much incentive to demand royalties or impose burdens on competing implementations. Some standards, of course, such as those for music and movie distributions on the web, are proprietary and licensed for a fee. Open source projects have typically refused to implement those encumbered standards, which may partly explain the dearth of infringement lawsuits. But when the stated goal of the standards setting organization is a royalty-free patent license for open source implementations, litigation over such patents is unnecessary.

The *Oracle v. Google* case is a recent exception. As I mentioned earlier in the copyright context, this lawsuit relates to Java software. Java is an industry standard for a popular programming language, and there are several proprietary and open source implementations of that language and its related programming libraries. Given the widespread reliance of the software industry on Java, it was a surprise when Oracle asserted a number of its patents (acquired in its acquisition of Sun)

against Google, purportedly for Google's implementation of the Android open source operating system.

It is far too early in this litigation to comment on the merits of the case, but the effect of the *Oracle v. Google* lawsuit has already been to put a cloud on the Java standard. This litigation is likely to discover some hitherto unexplored areas of software competition practices that are problematic given the explicit promises and community expectations of the parties to the Java Community Process under which Java was developed. In particular, the Apache Software Foundation has already complained about the JCP requirement that implementers acquire and pass a test compatibility kit (TCK) before receiving Java patent licenses from Oracle (Sun), when those TCK licenses expressly prohibit certain kinds of implementations and derivative works. That TCK licensing practice for industry standards is not compatible with open source, and those contractual restrictions on the use of specifications have not been accepted by Apache.

Traditional standards organizations have finessed the patent problem by requiring authors of specifications to offer reasonable and non-discriminatory (RAND) licenses. RAND doesn't mean "free", however, and it doesn't mean "without encumbering conditions," and so RAND promises are of little use to open source implementers. Fortunately, because open source implementations have become important as validations of open standards, most software standards are now actually published under RAND-Z, or zero-priced, RAND licensing terms. This promise of reasonable and non-discriminatory licensing terms is often made but not usually put into explicit words. For example, most contributions to IETF are accompanied by RAND promises, but actual licenses are not published anywhere on the IETF website. Nor does the W3C royalty-free patent policy require the publication of an actual patent license, although members of W3C are committed by their membership agreement to grant such a license if asked. Nobody asks.

Open source licensing practices no longer tolerate such ambiguities. Contributors to mature open source projects sign Contributor License Agreements to provide written confirmation of their copyright and patent promises. The Open Web Foundation (OWF) is drafting such explicit agreements for industry standards to mitigate the risks of patent litigation.

One important difference between the OWF proposal and traditional industry standard practices is in the identification of the patent claims being licensed for free. In most standards organizations the patent grant is to "Necessary Claims" – meaning those patent claims that are necessary to implement the specification without infringing. Actual terms differ; sometimes the rule is "actual technical impossibility" of implementation without a patent license and sometimes merely "financial or technical impracticality" of implementation unless a patent license is

available. Either way, this Necessary Claims language means that if there are multiple ways of implementing a specification, then no patent claim is a Necessary Claim.

OWF takes a more generous view of patent licensing for industry standards. Its patent grant says simply:

*The Promise. I, on behalf of myself and my successors in interest and assigns, irrevocably promise not to assert my Granted Claims against you for your Permitted Uses, subject to the following.*

The OWF agreements then define "Granted Claims" simply as those claims that are infringed by "Permitted Uses":

*"Permitted Uses" means making, using, selling, offering for sale, importing or distributing any implementation of the Specification 1) only to the extent it implements the Specification and 2) so long as all required portions of the Specification are implemented. Permitted Uses do not extend to any portion of an implementation that is not included in the Specification.*

This means that, so long as he is implementing the required portions of a specification, and even if there are multiple ways of implementing that specification, an implementer is free to choose a patented method if it is better or more efficient. The license extends only to the desirable goal of implementing the specification, however, and it is not a license for all uses of the Granted Claims.

To protect patent owners from unanticipated grants of their patents, as well as to protect implementers from game-playing by patent owners, the OWF agreements define Granted Claims as follows:

*"Granted Claims" are those patent claims that I own or control, including those patent claims I acquire or control after the Date below, that are infringed by Permitted Uses. Granted Claims include only those claims that are infringed by the implementation of any portions of the Specification where the Specification describes the functionality causing the infringement in detail and does not merely reference the functionality causing the infringement.*

Under the OWF, patent owners are thus able to read a Specification to determine which of their patents will be licensed to implementers, without needing to determine whether their patent claims are in some vague sense Necessary Claims because there are no alternatives. And implementers are able to read a Specification and, as long as they are implementing all the required portions of the Specification, they needn't worry about patent litigation for that implementation.

## **Conclusions and Predictions**

It is not yet known whether the OWF provisions for patent licensing will overcome the resistance to change for IP policies in standards organizations in order to make both copyrights and patents freely available to open source implementers of open standards. Nor is it known what effect the *Oracle v. Google* lawsuit will have on the Java standards and the future expectations of implementers to be free to create software based on open standards. Intellectual property attorneys live in interesting times.

In a way, this is very much like the challenges facing Creative Commons when it found chaos and uncertainty in the licensing of music and art and film for free use by all. Open standards are equally fundamental to the ways we live. That is why implementers ought to be free of copyright and patent restrictions to create the open source software on which our world depends. And that is also why attorneys should understand carefully the intellectual property obligations of contributors to and users of open standards.

---

<sup>†</sup> Lawrence Rosen is partner of Rosenlaw & Einschlag ([www.rosenlaw.com](http://www.rosenlaw.com)), a technology law firm that specializes in intellectual property protection, licensing and business transactions for technology companies. He currently advises many open source companies and non-profit open source projects, including as member of and counsel to Apache Software Foundation and as a member of the board of directors of Open Web Foundation. He is also serving at W3C on the Patents and Standards Interest Group and on the New Standards Task Force. Larry's book, *Open Source Licensing: Software Freedom and Intellectual Property Law*, was published by Prentice Hall in 2004. He is a Lecturer in Law at Stanford Law School.