

License Proliferation

by Lawrence Rosen¹

Emperor Joseph II: “Your work is ingenious. It's quality work. There are simply too many notes, that's all. Just cut a few and it will be perfect.”

Wolfgang Amadeus Mozart: “Which few did you have in mind, Majesty?”

Amadeus (1984)

Only a geek lawyer would compare the notes of a Mozart symphony to the cacophony of software licenses that is open source. That is because, as an author myself of several open source licenses, I “hear” some order and logic in licenses that is probably lost on most non-lawyers. So when folks start complaining about open source license proliferation I get defensive: Which licenses would you have us remove, Majesty, so that the symphony will be more pleasant to you?

As a license musician, I'm not bothered as much by “too many notes” as I am by the fact that the notes aren't always in the same key. License proliferation has become an important problem because software under those different licenses cannot always be played consistently and compatibly everywhere. Perhaps the Emperor means that we should throw out the off-key notes?

Imagine a world in which every word processing program created documents in its own internal format that could not be accessed directly by other word processing programs. This is not difficult to imagine. It is done on purpose even today by some proprietary software vendors—and it is enforced by their software licenses.

Similar things happen also with free and open source software: *Combining differently licensed software in useful ways is constrained by the terms of their licenses.* For example, the GPL says that “You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program...provided that you also meet all of these conditions....” The OSL says “Licensor grants You a license...to distribute...Derivative Works...with the proviso that....” Combining works under these two licenses is somehow constrained by the intersection of two different provisos.² Add to that other software under the Eclipse license, the MPL, the CPL, the Apache license,

¹ © Copyright 2005 Lawrence Rosen. Licensed under the Academic Free License version 3.0.

The author is founding partner of Rosenlaw & Einschlag (www.rosenlaw.com), a technology law firm with offices in Ukiah and Los Altos Hills, California. His firm advises many open source companies, contributors and users worldwide. Lawrence Rosen is the author of Open Source Licensing: Software Freedom and Intellectual Property Law (Prentice Hall 2004).

² A “proviso” is a legal term of art describing one or more conditions imposed on a party to a contract or license. Open source licenses impose a variety of conditions, some of which may not be acceptable to some licensees.

and the 60+ other licenses listed on OSI's website³, and it is no wonder the Emperor doesn't like the music.

The previous paragraph, by the way, itself contains an example of “off-key” legal writing too often found in open source licenses. I used the word *combining* as if it had legal significance. But it doesn't. The legal systems of the world won't know how to parse that word. Instead, copyright law uses the terms *collective work* and *derivative work*. Terms of art⁴ aren't always used correctly in software licenses, and so users aren't always certain what they can do—and what they must never do—with the software in their collections. Licenses that are so out of tune with the law perhaps ought to be discarded, or at least revised to bring them into tune.

Rather than just throw away some of the notes, though, I propose that we deal with license proliferation in a systematic way. Here are some suggestions:

1. Different expressions of the same concepts in different licenses should be justified on legal or business grounds, not just “freedom of expression by lawyers.” Identify where existing open source licenses say exactly the same thing in different ways. Pick one of those ways and agree as a community to stick with it for future licenses (or future versions of existing licenses). For example, the Grant of Copyright License provision can always include the proper language (at least in English!) to describe *all* rights under copyright in all Berne Convention countries. Differences in such provisions in open source licenses are unnecessary complications.
2. Agree to use legal terms of art properly, with proper respect for language and legal differences around the world. For example, where the terms *derivative work* or *distribution* are used improperly or ambiguously, they should be corrected or the situation documented.
3. Consistent with the goal of legal precision, encourage the creation of simple licenses. Eliminate “whereas” and “hereby” everywhere. Lawyerly writing needs to be read by human beings. Also be brief.
4. Agree on procedures to translate licenses from English into other languages. Start with the most important licenses—however “important” is defined by consensus—and then cooperate to ensure that translations are properly vetted. This process may itself

³ Open Source Initiative (OSI) at www.opensource.org/licenses.

⁴ Legal terms of art differ around the world and in some translations. That's another complexity of licensing law that license authors sometimes don't understand. Combining software under different licenses may involve untangling what different licenses in English say (or don't say) about, for example, whether dynamically linking two programs creates a derivative work. License authors have a responsibility to speak the law precisely.

discourage license proliferation, but in a natural way guided by actual needs around the world.

5. Licensors in the open source community should openly discuss and agree upon a standard Grant of Patent License provision. Patent law is tricky enough without subtle differences of license wording that may or may not allow licensees to create certain kinds of derivative works and may or may not require compliance with industry standards. We should decide whether we're licensing patents or patent claims, and express in standard ways the field of use and scope of our patent grants. Exceptions should be documented so nobody will be surprised by the complex patent grants in some licenses.
6. We should also discuss and agree upon standard patent defense provisions for open source licenses. Reasonable patent owners want to use their patents for defensive purposes against other patent owners. Meanwhile, licensees want to prevent over-reaching, where defense becomes offense against them.⁵ The open source community is now mature enough to be able to negotiate and compromise regarding patent defense provisions. The forum for that debate needs to be public—and it needs to be international. Exceptions to an agreed standard for patent defense in open source licenses should be documented.
7. We should decide what's reasonable for attribution purposes. The authors of open source software deserve recognition for their contributions; their reputations and their trademarks should be protected.⁶ On the other hand, free software and open source principles promise the freedom to create derivative works without having to place advertisements for the original authors all over software welcome screens. Can we compromise on attribution provisions, consistent with copyright and trademark law, that give credit where credit is due and software freedom elsewhere?
8. I recall law school classes in which we naïve students struggled over questions of jurisdictional diversity. What happens, I can now ask, when software licensed under the MPL (jurisdiction and venue in Santa Clara County, California) and software licensed under the CPL (jurisdiction and venue in New York) are combined? What if the

⁵ I intentionally use terms of military art. Patents are weapons. The subtle variations of wording of patent defense provisions in current open source licenses make it virtually impossible for most licensees to understand what's defensive and what's offensive about them.

⁶ In many countries there are laws concerning moral rights and author attribution. Since open source is a worldwide venture, we are eventually going to have to agree on common procedures for this, just as we have tried to do for such things as Internet domain names and trademarks.

person doing the combining is in Belgium or China? Surely we can compromise on reasonable jurisdiction and venue provisions in all open source licenses.

9. Understand and appreciate—indeed cherish—free and open source business model diversity. Software business models are expressed in software licenses. Some companies want their own licenses because they want to create their own limited commons of open source software from which derivative works can be made. Others are willing to license their patents only with broad patent defense provisions. Some companies and projects are, to put it simply, more generous than others. For a few important license authors, what matters most is their philosophy and its expression in the words of their license. License proliferation in this sense is a good thing. We ought not to prevent this or we will destroy the world of software freedom that we hope to create.

License proliferation when it happens carelessly or needlessly ought to be prevented. But we ought not to throw out notes just because there are too many of them. Instead, perhaps license authors can work together to achieve consensus about standard wording for standard licensing concepts. Simplicity and clarity, precision in our words, consistency whenever possible—these are solutions to license proliferation that will matter.