Tech Watch

Adriaan de Groot[a]

(a) Vice President of Legal Affairs, KDE e.V.

IFOSS L. Rev's Tech Watch section aims to allow technical experts and organisational leaders in Free and Open Source Software to introduce and explain topical issues with important legal aspects. These discussions may form the basis of subsequent detailed examination by IFOSS L. Rev. contributors.

This issue, KDE e.V. Vice President of Legal Affairs Adriaan de Groot reviews some of the issues presently confronting community software authors: copyright consolidation, making a living from coding, and 'doing legal stuff right'.

Keywords
Law; information technology; Free and Open Source Software; Bacula.org; Fiduciary Licence Agreement; Copyright; Copyright consolidation

Insofar as there is a "typical free software person," one might think of a software developer, the kind that sits up late writing code licensed as free software. Of course, here I mean free as in freedom to copy, modify, and distribute, although the real driving force for a developer may be the itch to create something useful or beautiful. At times like that, a developer's thoughts about legal issues are far, far away.

However, the real world intrudes in the end. Software needs to be used to be useful, and most developers want to distribute their software to as many users as possible. Distribution (or even assisting distribution, as in the Pirate Bay[1] case) means interacting with the rest of the world. One's private creative work is to be published with all the attendant details. Unlike many creative professionals, free software developers, especially the hobbyists, do not have agents to help with the legal issues and mechanics of distribution.

This journal is about the legal issues that attend free software development, publication and distribution in the broadest sense. It is a legal journal, largely from lawyers' perspectives. This column looks at what has been (or what should have been) keeping free software developers busy on the oft-neglected legal side of their work.

The biggest news on the copyright, patent and trademark front in the month of April 2009 is perhaps the initial verdict in the case of the Pirate Bay.[2] While it is of considerable importance to a large population of file-sharing enthusiasts, the impact on software developers and free software projects seems to be limited. One might argue about contributory infringement and how the development of software applications for file transfer can interact with the law. For the most part,

[1] http://www.piratebay.org

[2] Four found guilty in landmark Pirate Bay case, http://edition.cnn.com/2009/TECH/04/17/sweden.piracy.jail/index.html

though, free software developers are interested in working within the law, since it is that law -- and copyright in particular -- that makes free software licenses possible and necessary.

A developer does not exist in a vacuum, though. Assuming the software is interesting and useful to others as well, a community grows around that software. The community is more than only the developer -- there are users, writers, artists, system administrators and possibly even community managers (an odd job title, but one that flourishes in the free software world). "How do you eat?" is a common question, and securing the livelihood of all the members of a free software community is something that may require sound legal footing.

Two debates wandered around the free software world this month on topics directly related to legal matters. Both crossed the path of Tarus Balog, the lead developer of OpenNMS, a network monitoring suite released under the GNU GPL. OpenNMS is free software developed by a fairly large community, centrally managed and maintained, and supports a business. It is therefore a crossroads of forces around free software. The discussions spilled out across many different projects, especially after a discussion at the Open Source Business Conference[3] on community and licensing.

One debate is on the principles of licensing: what constitutes open source software? Where do paid-for services or code fit in to the free software world? Who controls the code? Where do the rights to the code reside? (These topics are culled from all of the blog entries that spawned around the original discussion; you might want to start at Aaron Seigo's blog post on copyright conslidation[4] or at Matthew Aslett's contribution[5], neither of which are by Tarus but which begin by examining the fringes of the argument).

The nature of discussion on the Internet between different communities is one of reaction-to-blog-entries, it seems. While communications within a single community are handled by mailing lists and forums, communication across community boundaries is mostly via public journals. As such it can be difficult to follow the discussions, and many red herrings can show up. At the tail end of the discussion (the first of the two blogs mentioned above), Aaron Seigo points out why fiduciary license agreements (FLA) can be good and useful, both to the community, organisations, individuals and corporations. The value of an FLA for a software project -- for the community, the organisation, individuals and corporations -- comes from the clarity it provides for long-term stability of the project. An FLA can be used to express the relationship between community contributors and the organisation or business that manages and maintains the project. That same FLA brings risks, though, since consolidating copyright in one place also provides a single point of failure.

The Free Software Foundation Europe and its Freedom Task Force have produced a model FLA[6] for use by Free Software projects. An early adopter was the Bacula project.[7] The KDE project, which produces the K Desktop Environment software (11 years, 6 million lines of code, 600 developers) has recently adopted its own version of that FLA.[8] Aaron's blog post reminds us why the KDE project recommends an FLA, but does not require one. Current efforts related to improving and extending the FLA include translation (on the FSFE's part) and standardised wording (for developers to fill in on their forms).

[3] Open Source Business Conference Retrospective, http://www.opensource.org/node/416
[4] http://aseigo.blogspot.com/2009/04/on-copyright-assignment.html
[5] http://blogs.the451group.com/opensource/2009/02/02/define-open-source-vendor/
[6] http://www.fsfe.org/projects/ftf/FLA.en.pdf
[7] http://www.bacula.org/fr/dev-manual/Fiduciary_License_Agreement.html
[8] http://ev.kde.org/resources/FLA-generic.pdf

For hobbyist developers, there is little interest in "doing it right" from a legal perspective. However, for long-term viability of free software projects, getting it right is something that needs to happen -- at least once. There is a great need for education, outreach and materials to enable free software projects to do the right thing by the law. Given the right tools, projects can engage with legal advisors to secure their own long-term viability.

About the author

__Adriaan de Groot__ is a researcher in software quality and verification; his free software efforts are confined to the KDE project, where he has the title Vice-President of Legal Affairs. His byline for developers is "I talk to lawyers so you don't have to." In this column, he writes for lawyers about what is going on in the trenches of free software development.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 1 (July 2009). It originally appeared online at http://www.ifosslr.org.

This article should be cited as follows:

de Groot, Adriaan (2009) 'Tech Watch', *International Free and Open Source Software Law Review, 1(1), pp 49 - 51*

Copyright © 2009 Adriaan de Groot.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.