

# Editorial

*Daniel M. German,<sup>a</sup>*

*(a) Associate Professor, Computer Science, University of  
Victoria, editor of this Review;*

DOI: [10.5033/ifosslr.v5i1.83](https://doi.org/10.5033/ifosslr.v5i1.83)

## **Abstract**

Editorial for Issue 1, Volume 5

## **Keywords**

Editorial

In Oct 1985, a few months after publishing his seminal GNU Manifesto, Richard Stallman founded the non-profit Free Software Foundation. It is likely that he had realized that he needed the legal framework of an organization that could own and administer the assets that he and his collaborators were creating, and that would manage the financial aspects of the project.

The communities of contributors of large Free and Open Source Software (FOSS) projects have found themselves in a position similar to that of Richard Stallman. As he did, they have created non-profit foundations to help them achieve their goals. The GNOME Foundation, the Linux Foundation, the Mozilla Foundation, the Apache Foundation, the Document Foundation, the Open Street Foundation and many others have been established by their corresponding communities to help manage their projects. In their paper titled “The Rise and Evolution of the Open Source Software Foundation,” Paula Hunter and Stephen Walli explore the reasons behind the creation of FOSS foundations from the legal, business, and technical point of view. They explain that, as projects grow in size and attract commercial interest, foundations are not only needed to manage the potentially conflicting interests of its participants, but to administer its assets and to help create a structure that supports and fosters the further development of its software.

Today, reuse is a very important aspect of software engineering. It is very rare to see a software product that has been developed completely from scratch. Instead, software is structured in layers, such that software systems “build” on the features of others. From a technical point of view, reuse is facilitated by well defined interfaces, typically known as Application Programming Interfaces or APIs. APIs become the protocol that defines how a software product (a library, an operating system, a programming language, a plugin, a web service, etc.) expects to interact with another one.

When the API that governs the interactions between two software systems is well-defined, either one of them can (at least in theory) be replaced by an alternative implementation that has the same API and equivalent functionality.

For this reason, it is important to know if a given API is copyright-able. If it is, then anybody wanting to implement a software system that implements such API would require a license from the API copyright owner. In this scenario, the API owner would be in the position to control the market of products that implement such API, potentially restricting competition. Perhaps more important is the question of whether APIs should be copyright-able at all. In a span of few months, two legal cases, one in Europe and one in the United States address this issue in a similar manner.

In Europe, *SAS Institute Inc. v. World Programming Ltd*, C-406/10<sup>1</sup> involved the SAS language. SAS is a programming language for data processing and statistical analysis. WPL created an implementation of this language without approval from SAS Institute, and without access to the source code of SAS. SAS Institute sued them for infringement of copyright.

In the United States, *Oracle America, Inc. v. Google, Inc.*<sup>2</sup> revolved around Java, the popular programming language, developed by Oracle America. Google had created, as part of its Android mobile platform, a partial implementation of the runtime library of Java – without the authorization of Oracle America, its copyright owner. Oracle America argued that Google had violated its copyrights, and sued.

Walter van Holst's article “Less may be more: copyleft, -right and the case law on APIs on both sides of the Atlantic” discusses both cases within the context of the licenses of the Free Software Foundation. In particular, he argues that if APIs are not copyright-able, then linking (whether dynamic or static) could be considered mere aggregation, and the General Public License could be interpreted to be equivalent to the Lesser General Public License (LGPL) and therefore weakened.

License proliferation is another issue that can potentially hurt software reuse in FOSS. One would expect that new FOSS licenses are created because their authors believe that current licenses do not satisfy their legal requirements. In the article “Copyleft: A Close Reading of the Lisp LGPL” Eli Greenbaum analyses the Lisp Lesser General Public License (LLGPL), a license derived from the LGPL version 2.1. Greenbaum describes the rationale behind its creation, and argues that the LLGPL is redundant and that its drafters would have achieved the same goals using the LGPL instead.

### ***About the author***

**Daniel M. German** is Associate Professor, Computer Science, University of Victoria. He completed his Ph.D. in computer science at the University of Waterloo. His work spans the areas of software evolution, software engineering of free and open source software and the impact of intellectual property in software engineering.

---

1 See *SAS Institute Inc. v. World Programming Ltd*. Case C-406/10 Judgement of the Court (Grand Chamber) of 2 May 2012. Available at <http://curia.europa.eu/juris/liste.jsf?num=C-406/10>

2 See *Oracle of America v. Google Inc.* Case No. C 10-03561 WHA. Order RE Copyrightability of Certain Replicated Elements of Java Application Programming Interface (N.D. Cal. July 22, 2011).

**Licence and Attribution**

This paper was published in the International Free and Open Source Software Law Review, Volume 5, Issue 1 (MARCH 2013). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

German, Daniel M (2013) 'Editorial', International Free and Open Source Software Law Review, 5(1), pp 1 – 4  
DOI: 10.5033/ifosslr.v5i2.83

Copyright © 2013 Daniel M. German.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND available at <http://creativecommons.org/licenses/by-nd/2.0/uk/>

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.



