

Intellectual Property Rights in Open Source Software Communities

Chitu Okoli

Concordia University, Canada

Kevin Carillo

Concordia University, Canada

INTRODUCTION

Intellectual property is an old concept, with the first recorded instances of patents (1449) and copyrights (1662) both occurring in England (“Intellectual property”, Wikipedia, 2004). The first piece of software was submitted for copyright to the United States Copyright Office in 1961, and was accepted as copyrightable under existing copyright law (Hollaar, 2002).

The open source movement has relied upon controversial intellectual property rights that are rooted in the overall history of software development (Lerner & Tirole, 2002; von Hippel & von Krogh, 2003). By defining specific legal mechanisms and designing various software licenses, the open source phenomenon has successfully proposed an alternative software development model whose approach to the concept of intellectual property is quite different from that taken by traditional proprietary software.

A separate article in this encyclopedia treats open source software communities in general as a type of virtual community. This article takes a historical approach to examining how the intellectual property rights that have protected free/open source software have contributed towards the formation and evolution of virtual communities whose central focus is software projects based on the open source model.

TOWARD THE OPEN SOURCE CONCEPT

With the advent of computer technology since the 1960s, communities of computer programmers began to form, starting with the MIT Artificial Intelligence Lab (AIL) in 1959. These “hackers”, as they called themselves, had a common culture that valued creative software solutions to various scientific and computing problems, and enshrined a culture where sharing ideas and programming code was expected and valued, with the most capable and inventive programmers receiving high respect.

However, things began to change in the early 1980s as computers became more ubiquitous, as physical sizes shrank and prices dropped while computing power simultaneously increased dramatically. Computing-based enterprises and even not-for-profit shops such as AIL began to realize the commercial value of software, and they started to enforce their copyrights and began to restrict sharing of software code strictly to their own organizations. Richard Stallman, a hacker at MIT’s AIL, opposed these moves to no avail. He finally quit in 1984 in protest against the restrictions on sharing among computer programmers, which he considered inimical to the hacker culture. He founded the Free Software Foundation (FSF) and with legal consultation created the concept of the “copyleft”, proclaimed in the GNU Manifesto (FSF, 1985) and legally enshrined in 1989 in the GPL (FSF, 1991).

Copyleft as expressed by the GPL has had a critical effect on shaping the very existence of open source software virtual communities. Open source software uses copyright law to preserve certain freedoms (hence the name, “free software”) regarding the creation, modification, and sharing of software. Specifically, all open source software grants users the following key rights:

1. **The right to full access to the source code:** When a computer programmer sees how a piece of software actually works, as specified in the source code, they can fully understand the inner workings and can intelligently modify the software as they deem appropriate.
2. **The right for anyone to run the program for any purpose without restriction:** There are no restrictions against commercial, military, foreign, or any other use, and discrimination against users for any reason is expressly forbidden.
3. **The right to modify the source code:** This includes absorbing the software, in whole or in part, into other pieces of software created by other developers.
4. **The right to distribute both the original software and the modified software:** A key difference be-

tween “free software” and “freeware” is that while freeware generally permits and encourages free distribution of the software, it does not permit sale of the distributed software beyond reasonable distribution costs.

5. **The right to know about their open source rights:** The open source license must be prominently displayed and distributed to users, so that they are aware of their rights (including access to the source code).

The GPL, the first legal document to license open source software, grants users and developers these rights with the intention that developers would modify the software and share it with others with similar liberality. This is a distinct concept beyond simple “open source” that the FSF calls “copyleft”. To guarantee this goal, the GPL grants the privileges mentioned above as long as a key condition is observed: *The obligation to distribute derivatives under copyleft*. Any software modified under the GPL can be redistributed for sale, but it must be licensed under a copyleft license; that is, modified derivative works must also be made available under an open source license. While it does not have to be licensed under the GPL itself, the chosen license may not restrict any of the five rights listed above.

These copyleft terms are critical to the very existence of OSS virtual communities. When Richard Stallman posted his manifesto and invited software developers to join him in his crusade for free software, there was no lack of sympathetic and willing hackers who wanted a return to the days of free sharing. However, there was a grave concern that, corporate interests could easily take these programs, add their proprietary extensions, and withdraw the software from public access. With its copyleft mechanism, the GPL guaranteed that any person or corporation who wanted to benefit from the liberal efforts of computer programmers would be legally bound to share their work in the same spirit of camaraderie. Considering the climate in which the free software movement was founded, it is unlikely that the movement could have gotten off the ground without such a radical clarion call to mobilize devoted followers in the first place.

IMPORTANT OPEN SOURCE SOFTWARE LICENSES, AND THEIR EFFECTS ON OPEN SOURCE SOFTWARE COMMUNITY LIFE

As detailed earlier, the GNU GPL was the first open source software license, and with its strong copyleft provisions, it enabled the possibility of open source software commu-

nities to form. One particularly strong feature of the GPL is its requirement that not only must derivatives of licensed software be copylefted (that is, made available under GPL-like terms), but all software programmatically linked together with GPL-licensed software must also be copylefted. This requirement, inspired by the Free Software Foundation’s stated goal of eventually ridding the world of proprietary software, has been widely considered excessive. In fact, no other organization has issued such restrictive open source software licenses. However, in spite of its strictness, the GPL remains the most popular licenses for open source software.

Based largely on the GPL, open source development communities such as SourceForge.net have flourished, protected by open source licenses that permit free creation and sharing of open source software. The most important addition to the GPL camp was Linux, which provided a long-sought kernel for the operating system being built by the GNU Project and that has now being proven to be powerful, fast, efficient, stable, reliable, and scalable (Edwards, 1998).

Loosening Up: Open Source Becomes More Commercial

In the 1990s, largely resulting from the phenomenal success of Linux, many of the organizations who had gradually commercialized their software in the 1970s and 80s came to appreciate the quality and quantity of work that could be done with their software when released to open source communities under the protection of appropriate licensing structures (West & Dedrick, 2001). However, few of these organizations felt comfortable with according rights as broad-sweeping as the GPL, and so gradually a wide variety of licenses were developed as various large software developers, both commercial and academic, began to experiment with releasing their source code for free development. These licenses avoided imposing the requirement of sharing such software under such rules; that is, they generally permitted developers to make proprietary derivatives from the selected source code they released.

Although the University of California already widely licensed their proprietary version of Unix, the Berkeley Software Distribution (BSD), they re-licensed it with the open source BSD License in the early 1990s (“Berkeley Software Distribution”, Wikipedia, 2004). The BSD license gives users the rights to run programs, to view and modify the source, and to distribute their modifications, including for commercial purposes. However, unlike the GPL, the BSD license does not require licensees to release the modifications by copyleft—they are free to make their modifications proprietary. Popular programs that use this

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/intellectual-property-rights-open-source/18087

Related Content

Network Effects

Erik den Hartigh (2008). *Encyclopedia of Networked and Virtual Organizations* (pp. 1007-1013).
www.irma-international.org/chapter/network-effects/17718

Building Trust in Virtual Communities

István Mezgar (2006). *Encyclopedia of Virtual Communities and Technologies* (pp. 4-9).
www.irma-international.org/chapter/building-trust-virtual-communities/18035

A Model for the Analysis of Virtual Teams

J. H. Erik Andriessen and Robert M. Verburg (2004). *Virtual and Collaborative Teams* (pp. 269-278).
www.irma-international.org/chapter/model-analysis-virtual-teams/30808

Sixth Sense Technology: Advances in HCI as We Approach 2020

Zeenat AlKassim and Nader Mohamed (2017). *International Journal of Virtual and Augmented Reality* (pp. 18-41).
www.irma-international.org/article/sixth-sense-technology/188479

Bunker-Room Mnemonics for Second-Language Vocabulary Recall

Alexia Larchen Costuchen, Larkin Cunningham and Juan Carlos Tordera Yllescas (2022). *International Journal of Virtual and Augmented Reality* (pp. 1-13).
www.irma-international.org/article/bunker-room-mnemonics-for-second-language-vocabulary-recall/304899