

Software Licensing in the University Environment **Katharine Ku**

The Intellectual Property Conundrum of Software

Is software more like a textbook, where the copyright is traditionally owned by the faculty author? Or is software more like an invention, whose idea is protectable by a patent? University faculty are not of like mind. Many electrical engineers who write code feel that software is just another means of performing a particular function, an extension of hardware. Many computer scientists think of code as a form of personal expression, much like words are to a writer. All would probably agree that the line between patents and copyrights is blurry when it comes to software.

Universities have been wrestling with the intellectual property conundrum of software. At Stanford University, we “cut the baby in half” by saying that patentable software falls under the Stanford’s Patent Policy and textbook-like software falls under Stanford’s Copyright Policy. In addition, if software has been developed with more than incidental use of university facilities (as stated in the Patent Policy), or with significant use of University resources (as in the Copyright Policy), or in the course of a research contract or grant from an external sponsor, the software will belong to Stanford. The licensing office must deal with interpretations of the policies in the individual context of the particular technology, using a reasonable and commonsense approach to decide whether it falls under the Patent Policy or the Copyright Policy.

How is ownership decided?

The Google and Yahoo stories illustrate the application of Stanford’s Patent and Copyright Policies to real-life examples. Jerry Yang and David Filo disclosed their software to Stanford, requesting that Stanford confirm that Stanford did not have an ownership interest in the technology. Yang and Filo were Ph.D. students at Stanford and had used Stanford computers (which is usually considered to be incidental use) to develop the software; their professors confirmed that their invention was not related to their university responsibilities as students. Based on this information, Stanford did not claim ownership to what became the Yahoo search engine.

In contrast, Sergey Brin and Larry Page had worked on a search engine for many years. Because the students had been paid by a government contract in the course of their research to satisfy their Ph.D. degree requirements, under both Stanford’s Patent and Copyright policies, Stanford had ownership to the software i.e., the written code. In addition, Stanford filed a patent on the method of ranking web pages in order to improve web searches. After trying to find the best licensee, Stanford determined that these inventors were in the best position to develop the invention effectively and so Stanford licensed the technology to their company, Google.

The technology in both cases were similar but the disposition of title to the intellectual property was different. Happily, both sets of students were able to exploit their invention, whether or not the University retained title. Thus, “ownership” in itself was not critical to the commercialization of the technology. Stanford benefited, even in the case of Yahoo, by the philanthropy of the successful entrepreneurs when they, two of the University’s youngest successful entrepreneurs, endowed the Yahoo Founder’s Chair in the School of Engineering.

To Patent or not to Patent: Does it make a Difference in Licensing?

Stanford has successfully licensed both patented and unpatented software. In fact, most of Stanford’s licensed software code is unpatented because it is often expensive to file software patent applications. On the other hand, we are willing to file patents on the ideas behind the software code if the concepts in the software are truly novel and enforceable and if we believe the royalty return will be worth the expense of filing.

An early example of patented and copyrighted software was the “Computer and Method for the Discrete Bracewell Transform”, filed in 1984 and issued in 1987, which described a particular mathematical calculation similar to, but more efficient than, the Discrete Fourier transform. Although Hewlett-Packard licensed the patent and included the technology in one of its products, the Discrete Bracewell Transform was not a significant commercial success.

MINOS, an unpatented linear and non-linear optimization program, has been licensed for 20 years to over 45 companies and non-profit entities in different fields of use; we consider MINOS a commercial success. The program is not patented, yet it is robust and has survived the test of time. Unpatented CAD/CAM programs such as SUPREM and PISCES have been successfully licensed to companies who have independently further developed the software for sale to their own customers.

Software is not just for the electronics industry. Genscan is currently one of Stanford's most popular unpatented software programs, licensed to almost 70 biotechnology companies who use the program to predict complete gene structures in genomic DNA sequences. The license is no longer negotiated on a case-by-case basis, but is readily available and downloadable on the web as a "ready to sign" agreement.

Exclusive or non-exclusive licensing or public domain

Stanford has had a long history of licensing software. In the 1980's, we recognized that software could be a valuable tool to researchers and that making these tools available to others would benefit the research community. OTL started a Software Distribution Center, which made nonexclusive licenses available "at cost" or at modest fees; OTL sent out tapes and handled the administrative aspects of distributing software. We now encourage software creators to make their software readily available to others for research purposes from their websites and maintain copyright protection via "click wrap" licenses.

We decide whether to license software non-exclusively or exclusively, depending on the stage of development of the particular software. If the software needs additional development before it can be widely used, we will often grant an exclusive license to give the company an incentive to invest the resources needed to commercialize the software. In the case of Google, the entrepreneurs were granted an exclusive license after extensive marketing to existing companies because Google was willing to devote significant resources to developing and improving the software.

In many cases, Stanford has licensed software non-exclusively to start-up companies for the sole purpose of providing the company "access", not a proprietary position, to the software. Stanford granted MIPS, founded by John Hennessy, then faculty member at Stanford and now President of Stanford, a non-exclusive license to use the software developed by Prof. Hennessy. Likewise, software developed by Prof. David Cheriton, was licensed non-exclusively to his start-up company, Granite Systems, which was subsequently acquired by Cisco Systems. Neither company needed an exclusive license to raise capital or develop products.

VXtreme was a company started by another then Stanford faculty, Prof. Anoop Gupta. Stanford did not file a patent but eventually granted VXtreme an exclusive license with a small amount of equity as partial consideration. VXtreme was subsequently sold to Microsoft and the University used the proceeds to help fund 25 Stanford Graduate Fellowships.

Lastly, if a creator/inventor wants to put her software in the public domain so that no one has any intellectual property rights in the software, or if a creator/inventor wants to make the IP freely available, Stanford will be agreeable, so long as such an action does not conflict with any existing contractual obligations and does not create any conflict of interest issue.

What is a Typical Royalty Rate?

The royalty rate for software can vary dramatically, depending on the stage of development of the product. If the software is readily commercializable, the royalty may be on the "higher side", as much as 20-25%, since the development cost will be minimal for the licensee. If the software requires extensive re-writing or development, the royalty rate may be on the "low side", perhaps in the 5% range. Some software is licensed for internal use only for a modest yearly payment. Financial terms of a license are typically negotiable, particularly for exclusive licenses, and are determined on a case-by-case basis.

Royalty Sharing

Most universities have a policy of sharing royalties with inventors and/or creators. But, the question of who should receive a share of royalty income for licensed software is sometimes a complicated issue. According to copyright law, a creator is someone who actually “writes” the code and so the programmer may be the appropriate royalty recipient of unpatented, copyrighted software. In patent law, an “inventor” is someone who “conceived an essential element of the invention”, thus an “inventor” of software would not have had to write a line of code but could be entitled to receive royalties. Lastly, some software is developed over several years and the code written by so many creators/authors that it is essentially impossible to know who all the creators have been. In such case, many universities would consider such a work to be an “institutional work” and would not share royalties with individuals. In most cases, royalty-sharing is determined on a case-by-case basis, based on policy guidelines.

Distance Education

The next frontier in the intellectual property debate about software is the area of distance education. For most universities, the intellectual property policies around “distance education” are not entirely established; the debate revolves around ownership of content. Many faculty feel that educational software is an extension of textbooks, which have traditionally been the property of faculty-authors. Many universities feel that the university resources often necessary to develop educational multimedia products entitle the university to some compensation or ownership in such educational material. This debate has been going on for many years and will likely to continue for many more.

Conclusion

University ownership in software is generally covered by a university’s copyright or patent policy. The disposition of title and royalties are complex issues and depend on the actual intellectual property protection for the particular software. Software creators should talk to an intellectual property professional for more information.