The Legal Fork In The OTD Roadmap – What Lies Ahead?

Bela Joshi, Ph.D.
Alion Science and Technology Corporation
Norfolk, VA
bjoshi@alionscience.com

Jesse D. Watters, Esq.
Alion Science and Technology Corporation
McLean, VA
jwatters@alionscience.com

ABSTRACT

In 2006, the Department of Defense (DoD) published a seminal document, the Open Technology Development (OTD) Roadmap, recommending the adoption of "open" technologies and practices within the DoD. The document advocates the adoption of open standards and interfaces, open source software, online collaborative tools, and technological agility in the acquisition and production of DoD software. The key motivation is to enable rapid deployment of the latest technology for the benefit of the warfighter.

There are multiple challenges associated with implementing OTD. Technical challenges are associated with developing software utilizing open source software, standards, design, and interfaces. Cultural challenges involve managing software processes for teams that may be geographically dispersed. Finally, there are legal challenges such as copyright, intellectual property, and licensing associated with the reuse of software components.

The technical and cultural challenges listed above are beginning to be well understood and have been documented through DoD case studies. However, the legal ramifications of adopting OTD have yet to be fully explored and understood. The OTD Roadmap itself provides little guidance on the impact of intellectual property, copyright, and distribution issues related to open software. This paper attempts to answer some of the common legal questions that arise with adoption of Open Technogies. We provide a high level overview of popular open source software licenses and describe terms and conditions associated with distributing software under these licenses. We also provide guidance on how to protect intellectual property and minimize the risk associated with the adoption of open source software.

ABOUT THE AUTHORS

Dr. Bela Joshi is a Senior Principal with Alion Science and Technology Corporation. Her interests include Agile methods for Software Engineering and Process Improvement. Dr. Joshi has an Electrical Engineering background and a Ph.D. in Engineering Management. She holds a Software Engineering Institute Certification in Capability Maturity Model Integration (CMMI).

Mr. Jesse D. Watters is Vice President and Associate General Counsel with Alion Science and Technology Corporation. He provides legal counsel related to contracts, data rights, ethics, and legal compliance. He served as Deputy General Counsel for the Coalition Provisional Authority, Office of the Inspector General and in various active duty and reserve commands over his 23 years in the U.S. Navy. He holds a J.D. from the American University and a B.A. in International Affairs from the George Washington University.

The Legal Fork In The OTD Roadmap – What Lies Ahead?

Bela Joshi, Ph.D.
Alion Science and Technology Corporation
Norfolk, VA
bjoshi@alionscience.com

Jesse D. Watters, Esq.
Alion Science and Technology Corporation
McLean, VA
jwatters@alionscience.com

INTRODUCTION - OPEN FOR BUSINESS

In recent years, the commercial sector has experienced tremendous growth in productivity and efficiency with the use of Open Source Software, Open Standards, and Open Interfaces. Wikipedia defines Open Source Software (OSS) as "computer software whose source code is available under a license that permits users to study, change, and improve the software, and to redistribute it in modified or unmodified form" (Wikipedia.org, 2007). Although the term "open" is often used in reference to software code, the term can apply to almost anything including standards, designs, interfaces, hardware, music, videos, documents, and text. There are many ways to designate these products as open, all based on the same fundamental concept, i.e., a product is open if the source materials can be used, modified, and redistributed by someone other than the creator.

OSS examples include Java, Apache, Eclipse, GNU, Subversion, MySQL, and of course Linux. Popular open projects are used by millions of people. For instance, the most widely used web server on the Internet is the Apache HTTP server with 58.56% market share (Netcraft Ltd., May 2007). SourceForge, an open source hosting site, has over 130,000 open source projects. In all likelihood, you have used open products in your daily work and may even be reading this with an open product. Open technologies are so pervasive in industry that "OSS technology stacks now form the basis of the bulk of Internet and information sharing technologies" (Scott et al, 2006). For an excellent survey that includes quantitative data on reliability, performance, security, and scalability of OSS, see David Wheeler's paper (Wheeler, 2007).

OPEN TECHNOLOGY DEVELOPMENT IN DoD

The extraordinary commercial success of OSS has not gone unnoticed within the Department of Defense (DoD). In 2006, the Deputy Under Secretary of Defense for Advanced Systems and Concepts (DUSD/AS&C) produced "The Open Technology Development (OTD) Roadmap Plan," a seminal document that advocates the adoption of OSS concepts within DoD. The document describes the urgent need for DoD to transform the way it does business by adopting open concepts, in order to rapidly serve the evolving needs of the warfighter (Scott et al, 2006).

Understandably, such a transformation implies a massive paradigm shift in the way the DoD acquisition, and vendor communities operate on a day-to-day basis. The roadmap identifies the challenges that have to be overcome as: Culture and Process, Software Project Governance, Software Policy, and Licensing and DoD Acquisition. Of these, the technical and cultural challenges are beginning to be well understood and have also been documented through DoD case studies (Joshi and Murphy, 2007). However, the software policy and licensing aspects affecting adoption are yet to be fully explored and understood. There is considerable confusion regarding the legal aspects that affect the implementation of OTD. Questions about choosing the right open source software, retaining intellectual property, maintaining open source license compliance, and using OTD for classified DoD projects abound.

This paper attempts to answer some of these questions by drawing upon the authors' experience as early OTD adopters as well as by surveying the OSS and legal landscape. We present a review of open source licenses and their applicability so that readers can understand the differences between various types of licenses. We identify some legal pitfalls associated with Intellectual Property (IP) infringement and copyright, and we present risk mitigation strategies and best practices in adopting OTD. It is our hope that raising awareness of these key business and legal issues will be beneficial as more and more DoD programs start to implement open technologies.

ALL ABOUT LICENSES

A software license is an agreement outlining various legal rights, duties, and obligations made between the creator/distributor of the software and the user of the software. It specifies a user's rights in terms of manner of use, access to source code, and ability to modify and distribute original and derivative works. An explanation of the licensing rights terminology follows. Use of software is defined as the right to use and run the software for any permitted purpose. Access to source code refers to the right to access the source code of the software in order to study or modify it. This is important, because if a user only has the compiled version and does not have access to the source code, he or she cannot modify and/or improve the software. The term "derived works" refers to modifications and improvements made to the original source code. Distribution is the right to sell or give away the modified software individually or as part of an aggregate software distribution containing programs from several different sources.

Licenses are used both in the open source software as well as the proprietary software domains. Both kinds of software licenses make use of the standard copyright mechanism to assert ownership in order to grant (or restrict) certain rights to the user. For example, while an open source license typically grants the rights to study, modify, and distribute the software, a proprietary license generally restricts such rights. Currently, over 50 licenses have been reviewed and approved by the Open Source Initiative (OSI), a public not for profit oversight organization whose charter includes the promotion of OSS and the approval of licenses to ensure they comply with the community norm for open source (http://opensource.org/). The OSI has published a standard definition for OSS licenses (http://opensource.org/docs/osd). The definition states that the essential feature an OSS license should address is the freedom to access the source code with the view to modify and redistribute under the same rights that are granted with the original software . The definition forbids discrimination against persons, groups or any specific domains of use. In addition, the definition states that a user should be free to extract a portion of the OSS and re-distribute with another product under the same terms and conditions.

Although all OSS licenses typically grant these freedoms to varying degrees, there are differences in the exact terms and conditions of the various open source licenses. Sometimes these differences are subtle, and hence it is important to obtain legal counsel to clarify the rights, duties, and obligations imposed by a particular license (American Bar Association, 2007).

Licenses can be broadly categorized as Strong Copyleft, Weak Copyleft, or Permissive. Copyleft, is a play on the word copyright. In proprietary licenses, a copyright is used to restrict freedoms or rights of a user. The copyright of OSS or copyleft licenses on the other hand is used to grant and guarantee these freedoms. A license is considered "strong copyleft" if it's terms and conditions are effectively imposed on **all** manners of derived works that originate from or use the original software, including for linking purposes. Strong copyleft licenses are sometimes referred to as viral licenses, because they "infect" software code, modules, or libraries that may be derived from the copyleft software, which in turn causes these derivative software products to themselves be copyleft when distributed. "Weak copyleft" refers to those licenses where the definition and scope of derived works that inherit the licensing excludes linking of OSS libraries. Permissive licenses, on the other hand, impose almost no restrictions on how derived works can be distributed.

GNU GPL

The Free Software Foundation (FSF) GNU General Public License (GNU GPL) is one of the oldest, and most widely used open source licenses (http://www.fsf.org). Linux as well as the popular free C++ compiler are released under the GNU GPL. The GNU GPL license authorizes a user to run, study, modify, and distribute the original software as long as the original copyright holder(s) are acknowledged. In addition, it requires all modifications or derived works to be distributed under the same terms and conditions defined by the GNU GPL, thus effectively guaranteeing that future derived works will continue to be distributed with the same freedom. Therefore, the freedoms granted by the GNU GPL cannot be alienated or removed from the original or modified source code. This means that if you combine GNU GPL code with proprietary code by linking the two together, the entire combined software package has to be distributed under the GNU GPL. While this protects the rights associated with the GPL, it does not guarantee or protect the rights associated with the proprietary code's license. Since software linked with the open source component is termed a derived work, regardless of whether modifications were made to the original code itself, the GNU GPL is considered very restrictive. The GNU GPL is referred to as a strong copyleft license because of its ability to impose copyleft restrictions on derivative works. (Note that the GNU GPL does allow restrictions for the use of compilers and Operating Systems).

The recently released GNU GPL version 3 addresses some of the concerns regarding the restrictive aspects of the GNU GPL (Smith, 2009). For example, GNU

GPL version 3 reduces the scope of the software affected by the terms of the older GNU GPL license, especially if no modifications have been made to the original open source software. The GNU GPL version 3 also addresses certain gaps in the previous license. It directly grants a patent license to the contributing organization and it addresses “outsourcing” rights to contracted companies.

LGPL

Other commonly used licenses differ from the GNU GPL in the manner in which derivative works are defined and the actual terms of their distribution. The FSF’s GNU Library or “Lesser” General Public License (LGPL) grants the freedoms of the GNU GPL but permits the linking of proprietary libraries without requiring that the proprietary libraries be distributed under LGPL as well. In this sense, the LGPL is a weak copyleft license. It is considered to be a weak copyleft license mainly because of the reduced scope for how derived works are defined and able to be distributed. This is important because a user is free to use un-modified OSS in their product distribution without having to make the source code of their product public. All the other freedoms associated with the GNU GPL, that is the ability to run, study, modify and redistribute still hold, as long as the terms of the LGPL are followed.

Apache

The Apache Software Foundation’s Apache License, like other OSS licenses, allows the user to use the software for any purpose as well as modify it with the view of distributing the modifications. Unlike the GNU GPL, it does not force a user to distribute the modified OSS software under the terms and conditions of the Apache License. The Apache License is thus also characterized as a weak copyleft license. The license does, however, require the user to attribute copyright by including the original authors and distributors.

BSD

The Berkeley Software Distribution (BSD) style license, which has its origins in the 1980s, is known for its brevity and simplicity. It grants all basic freedoms while also allowing developers to use its software to form the basis of proprietary software. A controversial advertising clause that made aggregating various open source software hard to achieve has been removed from the license. The BSD is therefore a permissive license.

MIT License

The MIT License is another commonly used a permissive license. It is very similar to the BSD license in that it permits the use of the OSS within proprietary software.

Public Domain and Government Produced Software

An essential criterion for a successful open source project is the license (Babcock, 2007). When a license has well understood terms and conditions, adoption by the user community is easier as there is no ambiguity as far as how the software can be used, modified and distributed. Software produced by government employees however presents a dilemma as far as a license is concerned. According to US Code Title 17 Article 105, software that is developed by government employees is not subject to copyright and falls under the “public domain”. This means that neither the government agency nor the government employee that produced it can claim copyright to the software.

The lack of a copyright and license presents challenge for the government. Since public domain code is not governed by any license, the government has no rights to source code if a third party provides subsequent modifications and improvements. Since licensing depends on a clear copyright, it is imperative to first establish copyright for government produced software. A recent DoD example illustrates how a third party can produce a derived work based on government produced software, in order to establish a copyright and a suitable licensing regime. Recently, DISA developed the Corporate Management Information System (CMIS), a Web-based workforce management and administrative software tool used to manage human resource, training, security, and acquisition. CMIS was released by DISA as public domain software. However, under a Cooperative Research and Development Agreement (CRADA) between DISA and Open Source Software Institute (OSSI) , OSSI produced a derived work based on CMIS software code. OSSI established a copyright and released the derived work as Open CMIS under the Open Software License (OSL) and the Academic Free License (AFL). A clear license will hopefully prompt commercial and academic organizations to adopt Open CMIS for internal use and provide source code containing enhancements and improvements in the future.

Open Source Licenses for DoD

The OTD Roadmap makes the case for creating a DoD/US Government Open Technology License that will meet DoD acquisition and policy requirements and be recognized in the commercial sector (Scott et al, 2006). The thinking is that a recognized license with defined and predictable terms and conditions will spur the rapid adoption of open technologies by DoD. By answering many questions regarding how an open source project can be used, linked, and re-distributed, the risks to programs using the licensed software would be minimized. A standard license thus would have the potential to help in the rapid dissemination and sharing of components among the different DoD programs, thus reducing the time it takes for new technologies to reach the warfighter. In addition, unnecessary rework on the part of DoD programs in creating individual release agreements will be eliminated.

Meanwhile, until this happens, special release agreements have to be crafted to enable distribution of the software to other government and DoD programs and agencies. The US Navy's Common Distributed Mission Training Station (CDMTS) is one such example (Joshi et al, 2006). CDMTS can be distributed under a special release agreement which states that recipients can only use the software for official government purposes, and the software may not be used for commercial purposes. In addition, modifications and improvements made by the recipients have to be approved by the Government. The agreement also protects the rights of developers to charge fees for providing modifications and enhancements as requested by the government.

GOOD COP – BAD COP: HOW ARE LICENSES ENFORCED?

While intellectual property is protected under the standard US copyright, patent and trademark laws, enforcing compliance with these laws is another matter. It is obvious that it is easier to steal IP in the open source world due to free and unlimited access to the source code. Generally speaking, it is up to the user community to "self-police" and report violations of open source licensing. According to Brett Smith of the FSF, when there are license violations, the preferred approach is to work with the responsible party with the aim of persuading them to stop such violations (Smith, 2008). Only when such tactics fail is recourse sought in the judicial system. This approach makes little sense in the corporate world, where profit is the motivator for producing software, and self-policing of license violations is not usually effective. However, the main goal of the open source community is to promote software accessibility so that software continues to be freely available in the public domain.

The recent May 2009 CISCO case provides a legal precedent in OSS license violation. CISCO released a popular wireless router with software that incorporated a GNU/Linux system. However, CISCO failed to distribute the source code with the product. In 2003, the FSF entered into discussions with CISCO with the aim of obtaining compliance (Smith, 2008). During the five years that followed, CISCO continued to release new products that were not in compliance with their open source license. CISCO defense was that source code was made available on their web site, or by written requests. However, source code that was made available was often incomplete and out of date, and written requests were simply ignored. This prompted FSF to finally sue CISCO in order to obtain license compliance. In the settlement that followed, CISCO agreed to employ a Free Software Director to supervise compliance with free software licenses and to periodically report on compliance status to FSF. Additional terms of the settlement were that CISCO agreed to notify previous customers of their rights under the GPL and other licenses, and also to publish complete software code on their website for download.

The CISCO case sets a legal precedent for violations of the copyright and distribution terms associated with OSS licenses. This was a major milestone for the FSF and for the future of the open source software community as a whole.

HOW CAN I MAKE MONEY IN THE OPEN SOURCE MODEL?

OSS is sometimes known as "free" software. Free OSS means that users of the software have the freedom to read/study, run, modify, and share software as well as derived works. However, free OSS does not necessarily mean the software is provided without cost. As long as distribution of the (derived) software is made in compliance with the license, organizations are permitted to charge money for services such as development, distribution, support, and maintenance of the software code. Even the strong copyleft GNU GPL explicitly makes provisions for an organization to distribute copies of the software licensed under it for a fee. As strange as this may sound, this is not only permitted but even encouraged by the FSF. There are several examples of commercial, for-profit organizations that provide support services to maintain or extend open source or free software. A well known example is the publicly traded Red Hat Inc., which has built a successful business around providing support, training, and integration services for OSS such as Linux. Hence, as long as all licensing terms and conditions are followed, you can still make money in the open source world. A successful example within the DoD modeling, simulation and training area is the LGPL licensed Delta 3D software, an open source 3D visualization and game engine (Joshi and Murphy, 2007).

WHAT ABOUT INTELLECTUAL PROPERTY?

IP is generally protected by copyright, patents, and trademarks associated with the software. The mechanisms used to protecting IP in the open source world are no different. For example, authorship and ownership can be asserted by inserting a copyright notice directly into each file of the software. Patents can be registered to grant the patent holder a temporary monopoly over the idea. A trademark in the form of a word, phrase, symbol, or design can also be registered to uniquely identify the producer of OSS. Copyright, trademark, and patent laws can be invoked to protect your IP in the open source world as well as in the world of proprietary software. How your IP is impacted by the use of or in combination with open source must be evaluated on a case by case basis by legal counsel to ensure these IP rights are appropriately preserved. Your IP developed while working in the open source model can be protected as long as you take the necessary legal steps to preserve your IP in accordance with the copyright, patent, and trademark laws.

Establishing IP via a copyright, trademark or patent is important, because once ownership is established then an organization can use the software in any manner. For example consider QT, a software product that provides a UI framework. It is released under a commercial license, LGPL and GPL version 3 (http://www.qtsoftware.com). The commercial license is meant for developers who do not wish to share the source code and comply with other terms of the LGPL and GNU GPL.

MY PROJECT IS CLASSIFIED. I CANNOT DO OTD, CORRECT?

Many DoD projects are classified. However, following sound software engineering design principles will allow the separation of the classified components and data from the rest of the system. Therefore, classified algorithms, business logic, and/or data can be isolated from the rest of the software modules. This enables the use of open source components for the non-classified portions of your system. Consider a project where you are required to build a Graphical User Interface (GUI) based system to allow users to access and manipulate classified information. Most likely, this project can be broken down into several modules or components. One such component may be devoted to user management in order to create users, assign roles and permissions and authenticate users. This component may involve a database, authentication, and UI widgets. The component outlined above does not require access to classified components or information that the final system will manipulate. Consult the definition of derived works and terms of distribution of the OSS license, before inserting OSS components into classified projects. As before, it is suggested that legal counsel be obtained for each individual project.

WHAT CAN I DO TO ENSURE COMPLIANCE?

There are many open source licenses that have subtle differences in the exact freedoms they grant to the OSS user. Consequently, it can be intimidating to navigate the legal and technical landscape in the open source world. This section provides high level guidelines to ensure compliance with open source licenses. Your organization should enlist the help of legal and technical experts to understand the terms and conditions of each license as well as how your system interacts with the open source software.

Several factors determine the impact of using and distributing OSS in your product. These are (1) the manner in which your organization's code interacts with the OSS, (2) whether the OSS itself is modified, and (3) whether or not the derived works are distributed externally. Each unique situation needs to be carefully evaluated against the terms and conditions of the OSS license.

Internal Use Only

If your organization uses the software for internal purposes exclusively, then you will not violate the terms and conditions generally associated with distribution to the outside world, such as making source code available. For example, consider the case where open source software forms the basis of your organization internal Payroll system. Even if you have modified and extended the OSS to customize for your own needs, as long as your organization does not distribute the code externally, the original terms and conditions of the OSS that apply to the distribution do not come into play. This applies equally to all OSS licenses, even the restrictive GNU GPL.

External Distribution

Now consider the case where your organization produces software for the purpose of distribution to customers (either with or without a fee). In this situation, there are three distinct ways in which your system can interact with the open source software.

Unmodified OSS Tools To Create Software: The first scenario is where your organization uses unmodified OSS to produce other proprietary or OSS software. Common examples would be using a development tool such as the Eclipse Integrated Development Environment (IDE) released under the Eclipse Public License, or an editing tool such as the GNU Emacs editor released under the GNU GPL. In this case, the OSS tools are used to create software that is distinct from the OSS tool itself. Since you are using the OSS in the standard unmodified manner and you are not distributing the OSS along with your product, there is no risk that your organization is violating any OSS license. This applies equally to any kind of OSS license, including the very restrictive GNU GPL.

Linking With Unmodified OSS: The second scenario is linking components of unmodified OSS with your own copyrighted code, be it proprietary or open source. Let's say that your organization produces proprietary software that links to an OSS via the original/standard interface, and these are to be distributed as an aggregate. In this case, as long as the OSS license is weak copyleft, source code for your proprietary software does not have to be distributed. Examples of weak copyleft licenses are Open Software License (OSL 3.0), Academic Free License (AFL 3.0), Apache License 2.0, Berkeley Software Distribution (BSD) License, Library, and Lesser GPL (LGPL). However, if the component that your proprietary software is linking to is licensed under a strong copyleft license (such as GNU GPL), you do have to make its source code available.

Modified OSS: Now consider the case where your organization needs to make modifications to the OSS in order to extend and improve it. If the OSS is released under any flavor of GNU GPL, your organization's proprietary source code as well as any modified OSS, has to be released. This is also true for most weak copyleft licenses. However, if the OSS is licensed under a permissive license, such as the BSD style license, you are not required to distribute source code modifications to your code or the OSS code.

In all cases, it is generally a good practice to acknowledge the copyright of the original OSS component(s), by inserting a notice in the source files or documentation.

HOW DO I MINIMIZE RISK?

First understand that if your organization employs software developers, it is likely that there is open source software incorporated in your code. Software engineers often research existing technologies or bodies of work before embarking on a new project or algorithm. Incorporating lessons learned by building upon existing software can allow increasingly scarce manpower and resources to focus on adding improvements and extensions instead of starting from scratch and reinventing the same ideas and concepts.

Due to the prevalence of open source software, it is important to establish guidelines and policies by enlisting technical as well as legal expertise. The following steps can help to minimize risk associated with open source in your project.

1. Define the software license for your own software.
2. Do not permit uncontrolled importation of OSS into projects. License terms should be cleared by the legal department prior to incorporation of open source components.
3. Perform analysis to understand how your product interacts with the open source components and how the open source software will be used. This is important to understand for license compliance.
4. Determine the types of open source licenses that are suitable for incorporation into your projects.
5. Formulate and implement an open source policy to prevent inadvertent incorporation of open source software. Too often, decisions to incorporate OSS are made at the individual software developer level.
6. Provide education for software engineers regarding open source licenses so that there is a common understanding of legal implications across the organization.
7. Large corporations may also wish to track and audit open source usage to ensure compliance against the open source policy by following the terms of the open source licenses incorporated in your product. You may also wish to enlist the help of a third party open source specialist to help your corporation comply.

It is clear that technical expertise as well as legal knowledge are required to make and enforce a good OSS policy in an organization.

CONCLUSIONS

The OTD movement is gaining momentum and the DoD is poised to take advantage of commercial style OSS and its phenomenal success. Adoption of OTD however, is a paradigm shift for the DoD acquisition and vendor communities. Questions about protecting intellectual property, licensing and distribution abound. Federal Acquisition Regulations and acquisition policies will have to be updated and adapted for DoD projects. Practical considerations and legal ramifications of using OSS will have to be evaluated on a case by case basis. Preferably, this activity should be undertaken early in the acquisition process, since licensing decisions impact not only the use and integration of the software but also the total life cycle costs of the system.

This paper attempts to provide some basic legal concepts and clarifications regarding OSS licenses. It is hoped that having a clear understanding of these issues will help to facilitate the adoption of OTD. As adoption of OSS increases and the OTD concept matures, legal precedents regarding copyright and licensing will be well established. As a result, the uncertainties and risks associated with OSS and OTD are expected to diminish as a clear legal and technical framework begins to emerge.

ACKNOWLEDGEMENTS

The authors wish to thank Curtiss Murphy for insightful discussions during the production of this paper.

REFERENCES

American Bar Association. “An Overview of ‘Open Source’ Licenses.” ABA Section of Intellectual Property Law American Bar Association. Accessed: 15 June 2009. <http://www.abanet.org/intelprop/opensource.html>.

Babcock, Charles (2007) What Makes The Cut? Information Week, Feb 5, 2007. <http://www.informationweek.com>.

Joshi, B., King, R., Teer, B. CDMTS: A Common User Interface for Multiple Training Environments, Proceedings of the 2006 Interservice/Industry Training, Simulation, and Education Conference (2006).

Joshi, B. and Murphy, C. “Are You Ready? The Open Technology Development Challenge.” Proceedings of the 2006 Interservice/Industry Training, Simulation, and Education Conference (2007).

Netcraft Ltd. “April 2007 Web Server Survey.” April 2007. Netcraft Ltd. Accessed: 10 June 2009. <http://news.netcraft.com/archives/2007/04/02/april_2007_web_server_survey.html>

Open Source Licenses by Category, Open Source Initiative, May 2008. <http://opensource.org/licenses/category> . Accessed: June 2009.

Open Source Software Institute, Program Overview 2007, http://oss-institute.org/OSSI/2007_OSSI_Program_Overview.pdf <www.oss-institute.org> Accessed: June 2009.

Scott, J., Lucas, M. and Herz, J. C. “Open Technology Development Roadmap Plan. Prepared for the Deputy Under Secretary of Defense ”Advanced Systems and Concepts.” (2006).

Smith, B., “More Bacground About the CISCO case,” 12-01-2008. Free Software Foundation, <http://www.fsf.org/blogs/licensing/2008-12-cisco-complaint>. Accessed: June 2009.

Smith, B., “A Quick Guide to GPLv3,” 01-08-2009. Free Software Foundation, <http://www.fsf.org/licensing/licenses/quick-guide-gplv3.html> . Accessed: June 2009.

Wheeler, D., Why Open Source Software / Free Software (OSS/FS, FLOSS or FOSS)? Look at the Numbers! 2007. <http://www.dwheeler.com/oss_fs_why.html> Accessed: 15 June 2009.

Weathersby, J. “Open Source Software and the Long Road to Sustainability within the U.S. DoD IT System“ Software Tech, Vol. 10, No 3. 2007. <https://www.softwaretechnews.com/stn_view.php?stn_id=42&article_id=85>