Open Archaeology: Fundamentals of Intellectual Property and Open Source

Carla Schroer

Cultural Heritage Imaging
A California Nonprofit Corporation
Topics

• What is Intellectual Property?
• Ownership and licenses
  – Open source, and open development
  – Main types of open source licenses
• Standards
• Factors to consider when choosing technology
• Content and media (images, documents, video etc.)
  – How do these differ from software?
  – Things to consider when choosing a license for your content
• Open discussion
Goals of the Workshop

• Help attendees:
  – understand factors in choosing software (open source and proprietary)
  – understand choices for licenses for their own work
  – find available resources

• Provide an opportunity for discussion and sharing of common concerns in this area
About Me

• 20 years in Silicon Valley software development

• 13 years at Sun Microsystems
  – Much of my work at Sun has involved licensing, including most recently the open sourcing of Java technology

• I am not a lawyer

• Co-founder of Cultural Heritage Imaging (www.c-h-i.org)
What is Intellectual Property? (IP)
Kinds of IP

• Copyrights
• Patents
• Trade Secrets
• Trademarks

Note: IP laws vary from country to country, though many concepts are the same
Most examples here are based on US Law
Copyrights

- Copyright - an exclusive right of reproduction or certain other uses of a literary, pictorial, audio or visual work.
- All software is subject to Copyright
- Can be assigned to another person or entity, or held jointly
- Generally the right to use a copyrighted work is granted through a license
“Copyrightability”

• **US (1976 law)**
  – Original authorship
  – Fixation
  – For software – *expression* of some functionality – not the functionality itself

• **EU (1991) Software Copyright Directive**
  – Created standards for software across the EU
  – Single Requirement – Originality

• Also includes derivative works
Derivative Works

• Work based on preexisting work
• For software – preexisting code
• Derivative work must contain an “original expression” of the later contributor
  – Size of the contribution is not relevant
  – Expression, not functionality matters
• Ideas are not copyrightable, only the expression of the idea is copyrightable
Patents

• Gives its owner monopoly power to exclude others from exploiting an invention
• May be licensed or bought and sold
• Infringement – in software means you implement everything included in one of the patent claims
• Independently coming up with the same idea does not protect you, unless you can prove that it preceded the patent holder’s invention (Timing varies in different jurisdictions)
Trade Secret

- Some sort of information that:
  - is not generally known to the relevant portion of the public
  - confers some economic benefit on its holder
  - is the subject of reasonable efforts to maintain its secrecy
- Is not protected once the secret is disclosed
- Does not protect against reverse engineering
- Companies protect trade secrets through non-disclosure agreements and employee contracts
Trademarks

• Identify the source of a product
• Can be a name and/or a logo
• “Strongest marks” have no other association, such as a made up word or an abstract design
• Fair Use – uses that don’t imply an official relationship or sponsorship,
  – Journalism and literature
Ownership and Licenses
Open Source

• A program is made available under an “OSI* approved” license which requires:
  – Free redistribution
  – Source code is made available
  – Allows for “derivative works” – downstream modifications
  – Other requirements

* OSI = Open Source Initiative
Open Development

• Open Source only means that the program and source code are made available under certain terms – says nothing about the development of the code

• Open Development means that the code is developed and maintained “in the open”:
  – People can view progress
  – Communities usually have rules for who can contribute, version numbers, source code control, etc.
Open Source Licenses

• Permissive licenses
  – BSD-Style, Apache

• Copyleft licenses
  – Mozilla Public License (MPL), GNU Lesser General Public License (LGPL), European Union Public License (EUPL)

• Strong Copyleft licenses
  – GNU General Public License (GPL)
Permissive Licenses

• Generally have these characteristics:
  – Unrestricted development of derivative works
  – Wholly unrestricted scope of license use
  – Can take derivative work and re-license under a different license, including commercial
  – May require attribution
  – May require limitation of liability on original author
Copyleft Licenses

• Unrestricted development of derivative works
• File-based licensing
• Files derived from a copyleft licensed commons must use the same license for source files
• Files not derived from copyleft licensed files may use any license
Strong Copyleft Licenses

- Unrestricted development of derivative works
- Project-based licensing
- Files derived from a strong copyleft commons must use the same license
- Under certain circumstances even files not derived from a strong copyleft commons must use the same license if they are distributed with code under that license.
Why Use Different Licenses?

• Permissive licenses
  – Allow the broadest use of the code with the least restrictions
  – Often used for sample code, and code the author wants others to adopt
  – Also used to create new markets

• Copyleft licenses
  – Ensure that work based on the licensed work remains open
  – Different “flavors” designate how strong the copyleft effect is
License Compatibility

- You cannot always use code from one open source license with code from another
- People often choose the same license as projects that are related to their work, or that they want to get code from, to avoid problems
- The Software Freedom Law Center and Free Software Foundation publish information on license compatibility
- There are “grey areas” and differences of opinion on what is allowed in some situations
- There are very few legal precedents
Standards
Open Standards

• A published standard that anyone may implement (caveat: patent rights)
  – Jpeg2000
  – HTML
  – Adobe pdf file format

• There may be both commercial products and open source products that implement the standard
Proprietary “Standards”

• Something a lot of people use, but there are no rights for others to implement it
  – Adobe Flash
  – Microsoft Office
  • There are some ways to get data in and out to other programs that are allowed, but the overall approach is proprietary
Weights
Money
Railroads
Screws and Threads
“When I use a word,” Humpty Dumpty said in a rather scornful tone, “it means just what I choose it to mean—neither more nor less.”

“The question is,” said Alice, “whether you can make words mean so many different things.”

“The question is,” said Humpty Dumpty, “which is to be master—that’s all.”
Why do Standards Matter?

- Allows sharing of content, metadata, and more
- Allows the possibility of usefulness in the future
  - How many of you can read “wordstar” files now?
  - How many of you have “PhotoCDs? Photoshop CS3 does not support that format
Open Standards Mean
No Vendor Lock-in
Choosing Technology
Considerations

• This discussion focuses on choosing technology for use in your projects, company, department, etc.

• It assumes that you are not planning to do development yourself to enhance the features, fix bugs, etc.
  – If you are planning to develop, then access to the source code is an important factor

• This is based on my own experience in the software business, and in running a small non-profit
Considerations

• What’s the technology for?
  – Do something (download pictures from a camera, run a scanner, transfer files from one place to another)
  – View something (web browsers, plug-ins for web browsers)
  – Create something (photos, documents, web pages, databases, 3D models, computer programs)
Considerations: Do Something

• Does it have the features you need?
• What are the costs? (including cost to maintain and cost to learn, not just cost to purchase)
• Is it reliable?
• Who is maintaining it? Will it be updated to support needed features (ex. new equipment, new operating systems, etc)
• Switching costs – what will it cost to switch to another solution, if this one no longer does what I need?
• Whether or not it’s open source is less important
Considerations: View Something

- Does it have the features you need?
- Does it properly view the file formats you care about?
- What are the costs?
- Who is maintaining it?
- Is it reliable?
- Whether or not it’s open source is less important
Considerations: Create Something

• Here is where open standards and open source play a big role
• Still have to consider questions about cost, reliability, switching costs, etc.
• What are you creating and who is it for?
Considerations: Create Something

• If you need to share it, you need to be able to import from and export to open standards

• If you care about people in the future having access to what you create, there are additional considerations
  – It is not just the software choice, but also the file formats you choose that affect usability
  – Ex Photoshop creates all of PSD, Tiff, DNG, JPEG2000, etc.
Considerations: Create Something

• Value of Open Source tools for creating Content
  – Open source based technologies make it easier for archives to migrate data to new systems - no dependencies on others
    • Does not solve all problems, but makes it easier
    • Depends on original source of code, how it’s written, how maintainable it is, etc.
  – Tradeoff between taking advantage of new approaches, and making sure things work in the future – open source can mitigate some of this concern
Creating and Licensing Content
Content

• Documents
• Photographs
• Video
• 3D models
• Drawings
• Animations
• Etc.
How is this Different?

• Open Source was developed for applications and operating systems, essentially “programs”
• Media and Content are not programs
• The essential ideas of “source code” and source code sharing do not readily apply
Licensing Content

- There is no one correct license
- Depends on what you want to allow and what you want to protect
- You do not need to license your content at all.
  - Claim copyright
  - Others cannot do anything with it, unless they get an explicit license from you
Creative Commons

- A family of licenses that allow you to choose certain restrictions/allowances:
  - Can it be used for commercial purposes?
  - Can it be modified?
  - Must it be attributed to you, and if so, what attribution is required? (i.e. company name, website link, etc)

- All CC licenses allow free redistribution of the material that is licensed.
Simple Content License Example

- Copyright © 2008 Cultural Heritage Imaging. Verbatim copying and distribution of this entire document is permitted in any medium, however, this notice must be preserved on all copies.

- Web site terms of Use
  - Claim copyright on content, and have terms of use for the web site which say what usage is allowed
Resources
Resources (1 of 3)

• SoftwareFreedom.org – Software Freedom Law Center
  – Particularly useful “resource” area, with free publications on open source licenses and copyright issues

• creativecommons.org
  – Great site, very easy to create your own license for a variety of materials
Resources (2 of 3)

- Wikipedia: Glossary of Legal terms in Computer Technology
- opensource.org - Open Source Initiative (OSI)
- archive.org – The Internet Archive
  - A place to find tons of content (texts, movies, audio, etc.) that is “open and free” also the home of the “wayback machine”
Resources (3 of 3)

• Open source communities:
  – fsf.org – Free Software Foundation - home of GNU software and creators of the GPL and LGPL
  – Apache.org – Apache Software Foundation home of the Apache community. Have their own software license, which is considered more “business friendly” than the GPL
  – Ubuntu.com – a community developed, Linux based operating system.
  – Sourceforge.com – open source software distribution
Acknowledgements

• Thanks to Patrick Curran of Sun Microsystems for use of his “standards” slides

• Thanks to the many lawyers I have worked with over the years who helped me understand this stuff. With special thanks to Chris Nadan, Jonathan Nimer, Tiki Dare, Melissa Mograss and Matt Marquis
Discussion