Copyright on Software. Legal Aspects of FLOSS

Legal Aspects – Master on Free Software 2012-13

Miguel Vidal

http://flossystems.com
Twitter: @mvidallopez

October 26th, 2012

FLOSSYSTEMS
Open Source Systems Engineering

Universidad
Rey Juan Carlos

Course Contents

- Lesson 1: Intellectual Property: basic concepts and legal framework
- Lesson 2: Copyright On Software. Legal Aspects of FLOSS
- Lesson 3: Free/Open Source software licenses
- Lesson 4: Free licenses for other intellectual works
- Lesson 5: Case studies

Table of Contents

1. Copyright on software
   - Origin, scope and reasons
   - Licenses
   - Public Domain
   - The Legal Framework for FLOSS Licenses
   - Types of FLOSS licenses
2. Software Patents
   - Software Patents
3. References

Copyright on software: History (1)

- Software came first as part of a hardware system (*bundling*)
- In 1969, IBM “unbundled” software and services from hardware sales (due to antitrust issues).
- Portable languages (C, Unix): software began to be distributed in an independent manner (1970s).
- At first, there was a big debate about if software should be protected by patents or by copyright.

Debate: Monopolies and Antitrust Laws

Debate: Monopolies and Antitrust Laws

- EU vs Microsoft: `http://goo.gl/XtHhK`

Debate: Monopolies and Antitrust Laws

- EU vs Microsoft: `http://goo.gl/XtHhK`
- What is a monopoly?
- Being successful is a monopoly?
- Single seller is a monopoly?
- Who can drive out (or limit) competitors of the market?
- Are specific Antitrust laws necessary?

Copyright on software: History (2)

The goals of the copyright on software were:

- Protect investments in the development
- Promote distribution of works
- Protect the creative human activity by providing incentives
- Protect a technology very prone to be copied

Copyright on software: Reasons

Copyright was finally chosen because of following characteristics:

- Simplicity (no registration, no formalities...)
- Automatic
- Inexpensive
- No novelty, just originality (it may be state of the art!).
- Includes documentation
- International (several conventions on copyright)
- Harmonization with other works.

Adapting the concept of copyright to software is not an easy task as there are many exceptions and special circumstances.

Copyright on software: Scope

What in software falls under copyright:

- The computer program (i.e. instructions, in any form): source code and object/binary code!
- The description of the program (for instance, its UML diagram)
- Additional material (user manuals, guides, etc.)
- Interfaces (graphics, sound, typographies...)
- Databases

Copyright on software: Boundaries

What in software leaves out of range of copyright:

- Algorithms
- Procedures
- Techniques used for development

Table of Contents

1. Copyright on software
   - Origin, scope and reasons
   - Licenses
   - Public Domain
   - The Legal Framework for FLOSS Licenses
   - Types of FLOSS licenses
2. Software Patents
   - Software Patents
3. References

Why Do I Need a License?

- Copyright covers source code.
- IP and Copyright is oriented toward preventing use of copyrighted material.
- If you don't license your code, it can't be used (legally) by other people!

No License Required?

- Copyright comes as soon as someone creates a "tangible" (expressible) work.
- In absence of any licensing declarations, don't allow any uses ("all rights reserved").
- Therefore, some declaration is necessary to allow sharing.
- One option is to declare no license is required to use the work (i.e. Public Domain Dedication).

Licenses and Communities

- Software licenses are social contracts just as much as they are legal documents.
- When you choose a license, you are charting a course for the future
- You are often establishing a relationship to a larger community.
- Not purely about mechanical and legal choices.
- It can be difficult change later: it is worthwhile spending time to understand it.

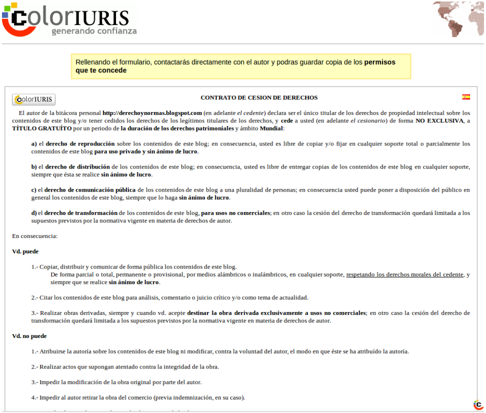
Licenses: Concepts

- An unilateral “contract” between the author and the user.
- Grants some rights to the users of copyrighted work.
- You don’t need to accept the conditions: but in that case, you don’t have any rights over the copyrighted work.
- You don’t need “to sign” the conditions (EULA is NOT necessary) Why?

Contracts and Licenses: Differing Views

- FLOSS licenses can be viewed as "bare licenses" or as contracts.
  - *A license is a contract:* (Raymond Nimmer and Van Lindberg) the contractual agreement is the essential factor.
  - *A "license" is NOT a contract* (Eben Moglen and Software Freedom Law center): it just exercises copyright.
- It's a tricky case to know if a particular agreement will be considered a bare license, a contract or both.
- "Pure license" interpretation make the enforcement of the free licenses much easier.

Licenses as a Contracts



Rellenando el formulario, contactarás directamente con el autor y podras guardar copia de los **permisos que te concede**

CONTRATO DE CESION DE DERECHOS

El autor de la bitácora personal **http://derechoynormas.blogspot.com** (en adelante *el cedente*) declara ser el único titular de los derechos de propiedad intelectual sobre los contenidos de este blog y/o tener cedidos los derechos de los legítimos titulares de los derechos, y **cede** a usted (en adelante *el cesionario*) de forma **NO EXCLUSIVA**, a **TÍTULO GRATUÍTO** por un periodo de **la duración de los derechos patrimoniales** y ámbito **Mundial**:

a) el **derecho de reproducción** sobre los contenidos de este blog; en consecuencia, usted es libre de copiar y/o fijar en cualquier soporte total o parcialmente los contenidos de este blog **para uso privado y sin ánimo de lucro**.

b) el **derecho de distribución** de los contenidos de este blog; en consecuencia, usted es libre de entregar copias de los contenidos de este blog en cualquier soporte, siempre que ésta se realice **sin ánimo de lucro**.

c) el **derecho de comunicación pública** de los contenidos de este blog a una pluralidad de personas; en consecuencia usted puede poner a disposición del público en general los contenidos de este blog, siempre que lo haga **sin ánimo de lucro**.

d) el **derecho de transformación** de los contenidos de este blog, **para usos no comerciales**; en otro caso la cesión del derecho de transformación quedará limitada a los supuestos previstos por la normativa vigente en materia de derechos de autor.

En consecuencia:

Vd. puede

1.- Copiar, distribuir y comunicar de forma pública los contenidos de este blog.
De forma parcial o total, permanente o provisional, por medios alámbricos o inalámbricos, en cualquier soporte, respetando los derechos morales del cedente, y siempre que se realice **sin ánimo de lucro**.

2.- Citar los contenidos de este blog para análisis, comentario o juicio crítico y/o como tema de actualidad.

3.- Realizar obras derivadas, siempre y cuando vd. acepte **destinar la obra derivada exclusivamente a usos no comerciales**; en otro caso la cesión del derecho de transformación quedará limitada a los supuestos previstos por la normativa vigente en materia de derechos de autor.

Vd. no puede

1.- Atribuirse la autoría sobre los contenidos de este blog ni modificar, contra la voluntad del autor, el modo en que éste se ha atribuído la autoría.

2.- Realizar actos que supongan atentado contra la integridad de la obra.

3.- Impedir la modificación de la obra original por parte del autor.

4.- Impedir al autor retirar la obra del comercio (previa indemnización, en su caso).

- http://goo.gl/tPcrl

Licenses: terminology

- licensor (“licenciante”).
- licensee (“beneficiario”)

Table of Contents

Public Domain

There are 4 common ways works arrive in the Public Domain:

1. the copyright has expired
2. the copyright owner failed to follow copyright renewal rules (ex. *Anthem*, Ayn Rand)
3. the copyright owner deliberately places it in the public domain, known as “dedication,” or
4. copyright law does not protect this type of work (facts, theories, short phrases...)

Public Domain: No Copyright

Sample Public Domain Dedication

"I, the copyright holder of this work, hereby release it into the public domain. This applies worldwide.

In case this is not legally possible, I grant any entity the right to use this work for any purpose, without any conditions, unless such conditions are required by law."

Public Domain Dedication

At the top of each file:

Sample Public Domain Dedication

The contents of this file are dedicated to the public domain. To the extent that dedication to public domain is not available, everyone is granted a worldwide, perpetual, royalty-free, non-exclusive license to exercise all rights associated with the contents of this file for any purpose whatsoever. No rights are reserved.

World copyright terms

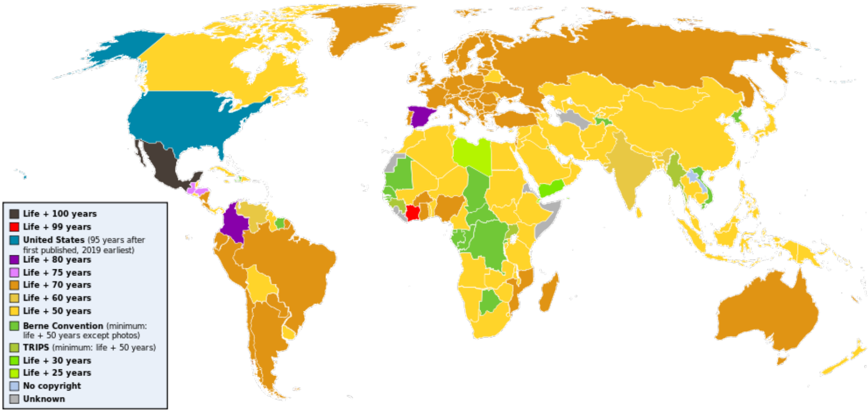


Figure: Worldwide map of copyright term length. *Source: Wikimedia Commons*

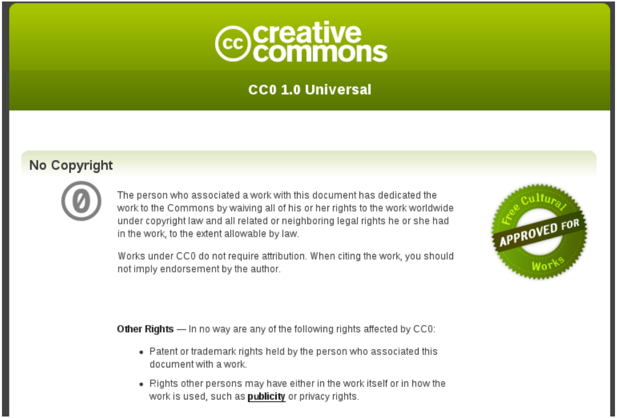
Public Domain Issues

- Is it possible in advance of the copyright expiration?

Public Domain Issues

- Is it possible in advance of the copyright expiration?
- Many legal systems effectively prohibit any attempt by these owners to surrender rights automatically conferred by law.
- Particularly moral rights (unwaivering, inalienable).
- A solution: CC0 (to waive all copyright to the fullest extent allowed by law).

Public Domain: CC0



creative commons

CC0 1.0 Universal

No Copyright

The person who associated a work with this document has dedicated the work to the Commons by waiving all of his or her rights to the work worldwide under copyright law and all related or neighboring legal rights he or she had in the work, to the extent allowable by law.

Works under CC0 do not require attribution. When citing the work, you should not imply endorsement by the author.

Other Rights — In no way are any of the following rights affected by CC0:

- Patent or trademark rights held by the person who associated this document with a work.
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights.

- http://creativecommons.org/publicdomain/zero/1.0/

Table of Contents

The Legal Framework for FLOSS Licenses

- Based on international copyright laws and provide the user with certain freedoms. These are granted as permissions which could not be exercised without the license (by default "all rights are reserved").
- Legal Hacking: FLOSS licenses behave as any other license except that they grant a number of rights to the user rather than restricting them.
- "Does this license apply in my country?"
- Question: Why law's changes are not necessary?

The Legal Framework for FLOSS Licenses (2)

Summing-up:

- FLOSS are consistent with IP laws: it's incorrect to suggest FLOSS licenses destroy IP.
- Legally, the only difference between proprietary and free software is the license (i.e. terms of use).
- Licenses (free or not) are based on every country's *copyright* law.

FLOSS License Example

To implement a basic free license is very easy:

Free License Example

Copyright (c) 2012 Foobar Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the redistributions of source code must retain the above copyright notice.

FLOSS License Example

To implement a basic free license is very easy:

Free License Example

Copyright (c) 2012 Foobar Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the redistributions of source code must retain the above copyright notice.

That's all!!

Should I write my own license?

Should I write my own license?

Why You Should Not Write Your Own License

Many people have attempted to write their FLOSS licenses but:

- You limit your community.
- You will probably get it wrong (ex. Artistic License 1.0... too clever!).
- Proliferation of licenses is harmful.
- Your code will not be Open Source (OSI) and (probably) not Free Software.

The Free Software Definition

FLOSS licenses are the mechanism to legally implement the 4 freedoms:

When you receive a free/open source software you get:

- Freedom 0 The freedom to use (run) the program, for any purpose.
- Freedom 1 The freedom to study how the program works, and adapt it to your needs.
- Freedom 2 The freedom to redistribute copies.
- Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

The Free Software Definition

FLOSS licenses are the mechanism to legally implement the 4 freedoms:

When you receive a free/open source software you get:

- Freedom 0 The freedom to use (run) the program, for any purpose.
- Freedom 1 The freedom to study how the program works, and adapt it to your needs.
- Freedom 2 The freedom to redistribute copies.
- Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

Freedoms 1 and 3 require access to a source code. All four freedoms must be granted at the same time!

Concepts related with FLOSS licenses

- Use: The right to use (run) the program, for any or some purposes.
- Redistribution: The act of copying the program and giving it to others.
- Derivative work: A program based in other program, reusing its source code.
- Authorship attribution: The obligation of recognizing the authorship of a work when applying any change, such as deriving or redistributing it.

The program is always owned by the license-holder. With the license, the user only get some rights of use ("economic rights").

Table of Contents

Types of FLOSS licenses

Every FLOSS license, no matter the kind of work, must guarantee the four freedoms mentioned above for the case of software.

Types of FLOSS licenses

Every FLOSS license, no matter the kind of work, must guarantee the four freedoms mentioned above for the case of software.

However, there are free software licenses more permissive and other more strict (the most strict licenses are known as “copyleft” licenses).

Types of FLOSS licenses

Every FLOSS license, no matter the kind of work, must guarantee the four freedoms mentioned above for the case of software.

However, there are free software licenses more permissive and other more strict (the most strict licenses are known as “copyleft” licenses).

Please note that two non-compatible free licenses doesn't imply that one of them is “less free” than the other.

FLOSS Licensing

From least to greatest complexity (and strict):

- Academic Licenses
- Permissive Licenses
- Partially Closable Licenses
- Reciprocal Licenses

Academic Licenses

- The simplest licenses: very few restrictions (close to PD).
- Reserving only attribution (keep names and copyright notice).
- Available for all uses, including use in proprietary closed source products.
- Originally written for and popularized by universities.

Academic Licenses

- The simplest licenses: very few restrictions (close to PD).
- Reserving only attribution (keep names and copyright notice).
- Available for all uses, including use in proprietary closed source products.
- Originally written for and popularized by universities.
- Examples: MIT, BSD, ISC.

Permissive Licenses

- Very similar to Academic Licenses
- Include grant of patent, trademark or public recognition provisions.
- Available for almost all uses, including use in proprietary closed source products.

Permissive Licenses

- Very similar to Academic Licenses
- Include grant of patent, trademark or public recognition provisions.
- Available for almost all uses, including use in proprietary closed source products.
- Examples: Apache License

Grant of Patent Licenses

Grant of Patent Licenses

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable, patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work , where such license applies.

Partially closable licenses

- Two simultaneous policies to the same code.
- Allow proprietary coders to reuse unmodified code as a whole (permissive-style)
- If there are any changes to code, it must be redistributed with the same license (reciprocal-style).
- Also known as: “weak copyleft” (or “reduced copyleft”).

Partially closable licenses

- Two simultaneous policies to the same code.
- Allow proprietary coders to reuse unmodified code as a whole (permissive-style)
- If there are any changes to code, it must be redistributed with the same license (reciprocal-style).
- Also known as: “weak copyleft” (or “reduced copyleft”).
- Examples: MPL, CDDL, LGPL

Reciprocal Licenses

- Code must allow others to freely and redistribute under the same reciprocal license.
- Also known as: "strong copyleft" (or "copyleft").
- Sometimes, "viral licenses": if reciprocally licensed code is incorporated, then the application is "infected" (the source code entire will remain under reciprocal license).
- It requires each binary distribution also include full source code.

Reciprocal Licenses

- Code must allow others to freely and redistribute under the same reciprocal license.
- Also known as: “strong copyleft” (or “copyleft”).
- Sometimes, “viral licenses”: if reciprocally licensed code is incorporated, then the application is “infected” (the source code entire will remain under reciprocal license).
- It requires each binary distribution also include full source code.
- Examples: GNU GPL, GNU Affero, Apple Public Source License (APSL)

What is the Copyleft?

What is the Copyleft?

- The FSF considered insufficient to grant the four freedoms mentioned above (use, copy, modification and redistribution).
- Copyleft makes sure that all users receiving a copy of the program get also the original four freedoms.
- It is an active defense of user's freedoms.
- The copyleft clause might have diverse implementations but all of them share the same concept: distribution of any version of this program must use this same license.

Types of FLOSS licenses

Free licenses can be classified in two main categories:

Types of FLOSS licenses

Free licenses can be classified in two main categories:

- Permissive licenses: The author retains copyright solely to disclaim warranty and require proper attribution of modified works, but permits redistribution and modification in any work, even proprietary ones (CC-by, *BSD, Apache, MIT).

Types of FLOSS licenses

Free licenses can be classified in two main categories:

- Permissive licenses: The author retains copyright solely to disclaim warranty and require proper attribution of modified works, but permits redistribution and modification in any work, even proprietary ones (CC-by, *BSD, Apache, MIT).
- Copyleft licenses: The author retains copyright and permits redistribution and modification provided all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under the same 'copyleft' license. Also known as “reciprocal licenses” or “share-alike” (GPL, GFDL, CDDL, CC-by-sa).

Types of FLOSS licenses

Free licenses can be classified in two main categories:

- Permissive licenses: The author retains copyright solely to disclaim warranty and require proper attribution of modified works, but permits redistribution and modification in any work, even proprietary ones (CC-by, *BSD, Apache, MIT).
- Copyleft licenses: The author retains copyright and permits redistribution and modification provided all such redistribution is licensed under the same license. Additions and modifications by others must also be licensed under the same 'copyleft' license. Also known as “reciprocal licenses” or “share-alike” (GPL, GFDL, CDDL, CC-by-sa).

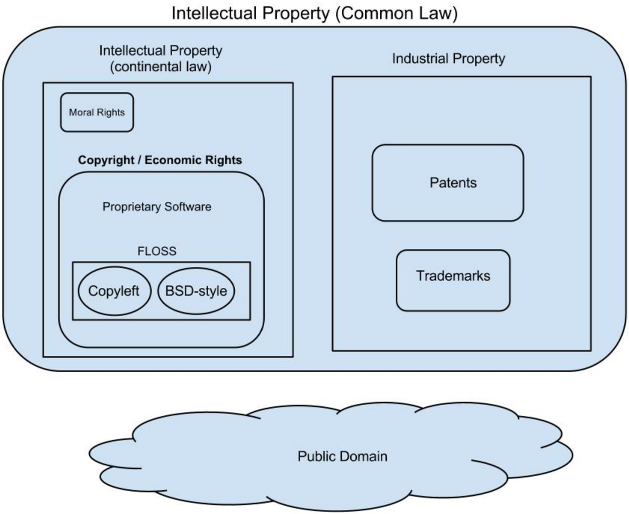
Please note that both license types are for free software. But with the first type, you can do proprietary derivative works, and with the copyleft license not.

Exercise: Venn Diagram

Show logical relations between these concepts:

- Copyright
- Patents
- Intellectual Property (continental law)
- Trademarks
- Free/Open Source Software
- Public Domain
- Copylefted software
- BSD-style software
- Moral Rights
- Industrial Property
- Proprietary Software
- Intellectual Property (common law)
- Economic Rights

IP Concepts


Intellectual Property (Common Law)
Intellectual Property (continental law)
Moral Rights
Copyright / Economic Rights
Proprietary Software
FLOSS
Copyleft
BSD-style
Industrial Property
Patents
Trademarks
Public Domain

Restrictions and FLOSS

- Are there permissible restrictions in FLOSS licenses?

Permissible restrictions

- Attribution of authors (such attribution does not impede normal use of the work).
- Transmission of freedoms (copyleft or reciprocity).
- Protection of freedoms (access to source code or prohibition of “technical measures”, DRM).

Warranty and disclaimer

- Software by itself is not a consumer product.
- When software is (combined into) a consumer product, disclaimers are ineffective.
- "As Is": we are accepting item in the actual state **with all faults**.

BSD Warranty Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License compatibility

- Two licenses are compatible if a joint derivate project could be delivered (i.e., the resulting code can be redistributed together).
- Compatibility is determined by comparing restrictions imposed by each license.

Compatibility == merge source code from different FLOSS software licenses.

Dual-licensing

Distribute software under two different sets of terms and conditions. Motivations:

- License compatibility (Perl, Mozilla/Firefox, MySQL).
- Market segregation based business models (MySQL Enterprise)
- Allows the holder to offer customizations, early releases, generate other derivative works or grant rights to third parties to redistribute proprietary versions.

Proliferation of licenses

- Vanity licenses: It has been a known problem in the community for a few years.
- A growing number of licenses increases exponentially the possible combinations and interactions.
- This fact makes it difficult to merge code from diverse sources, both for incompatibility issues and unacceptable clauses.
- It introduces juridical insecurity requiring lawyers, that it is what free licenses where trying to avoid in the first place (i.e. the EUPL license and EULAs).
- It favors FUD (Fear, Uncertainly, Doubt).

Table of Contents

1. Copyright on software
   - Origin, scope and reasons
   - Licenses
   - Public Domain
   - The Legal Framework for FLOSS Licenses
   - Types of FLOSS licenses
2. Software Patents
   - Software Patents
3. References

Patents. Concepts

- Designed to protect inventions and technological developments.
- Exclusive rights (monopoly) granted by a sovereign state to an inventor.
- Limited period of time (20 years) in exchange for the public disclosure of an invention.
- Certain subject areas can be excluded: business methods and computer programs.
- Philosophical, mathematical or scientific discoveries cannot be protected.

Patents. Inventions

Designed (theoretically) to be hard to get:

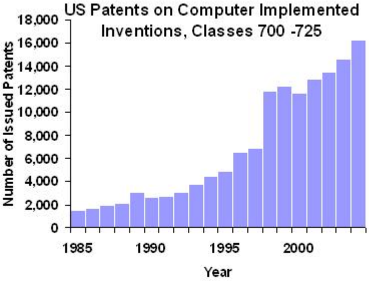
- Useful: Industrial devices or processes that perform a practical function.
- Novelty: no “prior art” (an idea that isn’t already in use).
- Non-obviousness.

Patents

- Cost from $10K to $50K to obtain.
- Require assistance of a patent lawyer.
- Patents protects all implementations of a particular idea.
- Right to exclude other people from making or using patented invention.
- Even if they didn't copy the idea! (independently invented or re-implemented).

Software patents

- Definition disputed.
- Intense and active debate for years.
- European Union: article 52 (European Patent Convention) excludes “programs for computers as such”.
- Japan: Software-related inventions are patentable.
- USA: software patents allowed.



Software patents

- Europe: Although not legal, in practice, lots of algorithms, and in fact, ideas, have been patented.
- Since trivial ideas (implemented with algorithms) are patented, they are often used by owners to drown competitors.
- It is very easy to infringe a lot of patents when developing a software project.
- "Patent trolls": patent owner who doesn't manufacture or use the patented invention, but it seeks to enforce its right through the negotiation of licenses and litigation. Also known as Non-Practicing Entity (NPE).
- PatentFreedom.com provides updated information about patent trolls and NPEs.

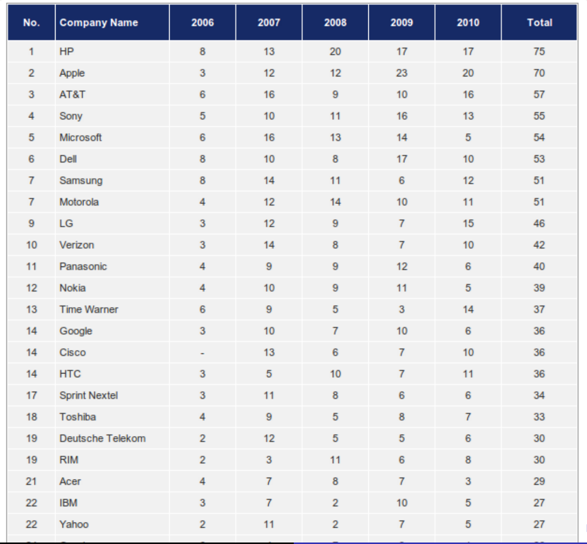
Examples of Software Patents

- GIF patent ('302 US Patent, 1987)
  - LZW compression alghorithm.
  - Unisys (1999) announced anyone using GIFs would have to pay 5000−7000.
- 1-click Amazon:
  - Trivial: to make online purchases with a single click!
  - Reexamination in 2006 (prior art).
- XOR cursor: it's visible on black and white surface.
- RSA patent (cryptography).

A patent-breaking brain

- Short tutorial on mentally performing the LZW method on a 1-pixel GIF.
- After completing tutorial, site visitor was informed his brain infringed GIF patent.
- To continue thinking the visitor should be prepared to pay the license fee.
- http://lists.tunes.org/archives/cybernethics/1999-September/000057.html

Patent trolls - NPE



No.	Company Name	2006	2007	2008	2009	2010	Total
1	HP	8	13	20	17	17	75
2	Apple	3	12	12	23	20	70
3	AT&T	6	16	9	10	16	57
4	Sony	5	10	11	16	13	55
5	Microsoft	6	16	13	14	5	54
6	Dell	8	10	8	17	10	53
7	Samsung	8	14	11	6	12	51
7	Motorola	4	12	14	10	11	51
9	LG	3	12	9	7	15	46
10	Verizon	3	14	8	7	10	42
11	Panasonic	4	9	9	12	6	40
12	Nokia	4	10	9	11	5	39
13	Time Warner	6	9	5	3	14	37
14	Google	3	10	7	10	6	36
14	Cisco	-	13	6	7	10	36
14	HTC	3	5	10	7	11	36
17	Sprint Nextel	3	11	8	6	6	34
18	Toshiba	4	9	5	8	7	33
19	Deutsche Telekom	2	12	5	5	6	30
19	RIM	2	3	11	6	8	30
21	Acer	4	7	8	7	3	29
22	IBM	3	7	2	10	5	27
22	Yahoo	2	11	2	7	5	27

Patents: Incentive for software innovation?

A reason for its existence is that it encourages inventions to be shared rather than be kept secret. Is it true with software?

Patents: Incentive for software innovation?

A reason for its existence is that it encourages inventions to be shared rather than be kept secret. Is it true with software?

- The academic community publishes their innovations to the public.
- There is a massive and rapidly growing amount of innovative open source software.
- Companies have strong incentives to participate in open source.

Software patents

- The patent system doesn't exist to protect intellectual property.
- The patent system exists to provide an incentive for innovation *where that incentive would not have existed otherwise*.
- If the incentive to create the innovation was there without the patent system, then the patent system is serving no purpose.

Software patents

Do patents encourage innovation in startups by protecting them from having their ideas "stolen"?

Software patents

Do patents encourage innovation in startups by protecting them from having their ideas “stolen”?

- Patents are a cost.
- Startups must build “defensive patent portfolios”.

References

- Van Lindberg, *Intellectual Property and Open Source*, O'Reilly, July 2008.
- Malcolm Bain et al. *Aspectos legales y de explotación del software libre*, UOC, February 2007. `http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/aspectos-legales-y-de-explotacion-del-software-libre/materiales/`
- Lawrence Rose, *Open Source Licensing*, Prentice Hall, July 2004