



FLOSS and the legal landscape

OSM2011
Kaido Kikkas

What is it all about

- On one hand, free licenses stem from the same legal space as the proprietary ones (e.g. GPL is based on the concept of copyright)
- On the other hand, they have a radically different approach to the application of the rights
- **DISCLAIMER:** the lecturer is extremely critical about the current situation of the so-called intellectual (some say “imaginary” instead) property

Do we need to care...?

- Isn't it just some lawyer stuff?
- A sample case from the real world from 2007 – Estonian EU officials in Brussels had legal problems with watching Estonian TV. They had bought the equipment in Tallinn, making it illegal to use there. And yes, it can go much more idiotic than that
- Before going to licenses, we need to study some background

Sometime, somewhere...

- ...something went terribly wrong
- Yet the original idea of IP was positive in the context of the time; the Queen Anne Statute was a remarkable legal document
- Ct. Leonardo: Do not teach anything and you alone will excel
- Old Man Paragraph vs the Internet Kid
- Mindquake (Robert Theobald)

The traditional model: WIPO

- Established in Stockholm 1967
- Two goals:
 - on one hand, to provide fair compensation for creators
 - on the other hand, to allow the society to use the creation
- defines intellectual property as any result of human intellectual activity (ideas, inventions, scientific theories, works of art etc)

IP according to WIPO

- Shares many qualities with physical property:
 - can be rented, sold and bought
 - can be given away or stolen
- Specific features:
 - immaterial
 - territorial
 - temporary

Main categories

- Copyright (+ related/neighbouring rights)
- Industrial property
 - patent (+ utility model rights)
 - trademark
 - trade secret
 - industrial design
- Competition laws



Copyright vs industrial property

- The Object
 - art, literature and science vs industry
- The Owner
 - author vs applicant
 - application system vs author system
- The Rights
 - automatic vs set procedure
- The Duration
 - eternal/life+70 vs shorter durations

Copyright

- Besides proprietary approaches, also free licenses stem from copyright. Yet, the traditional approach has a number of serious problems:
 - Unreasonable durations
 - Rabbit's Friends and Relatives (“Winnie the Pooh”)
 - Jack O'All-Trades, Certified Dentist and Gas Welder
 - Old Man Paragraph vs the Internet Kid
- Also, the Anglo-American (US, UK) *copyright* and Continental European *droit d'auteur* are quite different (fair use, selling the rights etc)

Patent

- Latin 'patens' = public; main idea: publish and compensate instead of keeping secret
- mostly used for inventions: "a new way to create something old, or vice versa"
- Patenting criteria:
 - novelty - differs enough from the existing things
 - grade of invention - must be nontrivial
 - usability - can be produced/used in larger scale

Patentable or not

- Can be:

- devices
- methods
- substances (including biological)
- the combinations of the previous

- Cannot be:

- design solutions
- animal breeds
- scientific discoveries
- algorithms and computer programs

Software and patents

- Both common sense and earlier concepts exclude software from being patentable (previous slide)
- Yet, in the US the large firms managed to change the laws at the end of the 20th century
- Europe has no software patents (yet)
- What it might mean: <http://webshop.ffii.org>

License

- A legal permit to use someone else's creation (IP) under predetermined conditions
- Can be limited by
 - ways
 - purposes
 - period
 - territory
 - extent or instrument

Problems with software licensing

- Most licenses (nearly all proprietary and many free ones as well) are incomprehensible for normal people. How many of you have read the license of your Windows?
- Many proprietary licenses (EULAs) have grown unreasonable – e.g. trouble with hardware changes or moving to another machine

Why it does not work –

1. greedy deadlines

- The original U.S. copyright was registered for 14 years, later became possible to extend for 14 more
- Today, most copyrights extend to 70-95 years **after the author's death** – will create a new social group of pensioners whose sole achievement is having a right grandfather
- While 20 years of patent might have been adequate when James Watt built his steam engine, it is ridiculously overextended by today's standards, making nearly all technology a legal minefield



Why it does not work –

2. Rabbit's Friends and Relatives

- The problem is with the neighbour rights – in addition to just e.g. composer, lyricist and singer as beneficiaries, there are n+23 Third Assistants of 2nd Manager (or Lawyer). And everybody wants a piece of cake
- Example: most musicians' income goes to the recording corporation (some exceptions: Queen, Metallica)
- The Switchfoot DRM case: “Sorry guys, we did not order it!”



Why it does not work – 3. Jack O'All-Trades, CDW*

- There are
 - IT and new media people
 - Lawyers
 - To make at least somewhat intelligent decisions in “protecting IP”, one must be competent in BOTH fields – and these people are extremely rare
 - Result – a lot of decisions go wide: patents cover adjacent things which should never be covered
- * Certified Dentist and Gas Welder

Why it does not work –

4. Old Man Paragraph vs the Internet Kid

- Fundamental contradiction between the dynamic world of technology and the conservative, static legislation
 - A good law will not change frequently
 - Stagnant technology is a Bad Thing
- How to put the static legislation in charge of the dynamic technology?
- Example: Software is still treated as works of literature under the Berne Convention (1979)

In days of old...

- ... software was not a product - licenses were unnecessary, there was no place for intellectual property. Reasons:
 - an elite thing for the elect few
 - military connections
 - bureaucracy was kept separate
 - resources were scarce
 - hardware-specific software

Proprietary software

- software with restrictions imposed by its proprietor (owner) on its use, copying and modification:
 - legal
 - technical
 - both
- NB! The stress on different measures has changed over time

Types of proprietary software

- As it is often said but still frequently misunderstood - "proprietary software" does not mean the same as "commercial software"
- All free software can be used commercially under their licenses, while there are kinds of proprietary software that cannot (e.g. "free for non-commercial use" freeware)

Commercial proprietary

- Sold commercially by unit ("boxed" software; recently sometimes also by authorised download)
- a very limited set of rights - specific computer and often also specific user
- Copying is very limited (often, users can make a single copy for backup; other copying is forbidden), reverse-engineering, closer studying, modifying and deriving new products is prohibited

Shareware

- Proprietary software which can be freely copied, shared with others and also used for a limited period
- Does not allow modifications and is distributed as closed-source
 - adware
 - nagware
 - crippleware
- Becomes illegal to use after expiration of the trial period (protected by legal means)

Freeware

- This is proprietary software with zero price
- Freeware < > free software
- Freely copied and used - however, many products limit the use by purpose, the most common restriction being "free for home or non-commercial use"
- Regardless of use, reverse-engineering, studying and modifying is forbidden, just as developing new products

Public domain

- Not proprietary (one of the most permissive legal categories), but can be used in proprietary context
 - binary only
 - closed derivatives
- Even some free licenses allow for proprietary derivatives (e.g. BSD, MIT or X11 license). This kind of activity is impossible with free software licenses with "strong copyleft" that forbids narrowing the users' rights

Special notion: Microsoft Shared Source

- Microsoft's campaign from around 2003
- Touted as a counterpart to Open Source
- A family of special license agreements with certain partners (countries and large companies) who have been given partial access to some of the proprietary code
- Bruce Perens: "look but don't touch - and we control everything"

Arguments for...

- “Innovation is driven more quickly when it is lucrative - the best way to ensure this motivation is to tie revenue to innovation”. Definitely not an universal principle, lots of examples
- “Proprietary software is said to create greater commercial activity over free software, especially in regard to market revenues”. Ignores the fact that the business models are different
- “Ordinary users do not care - they want a simple and working solution”. Somewhat true – however, they will start caring if intrusions get too big (WGA)

... and against

- Trends of greed
- Vendor lock-in
- Pig-in-a-bag
- Privacy issues
- Abandonware

Where it might make sense

- High-level, non-IT professionals (architects, composers...) needing turnkey solutions:
 - very specific skills
 - little or no IT competence
 - time is money
- But even here, a free solution might be preferable:
 - the client has no difference
 - the technical maintainer has got larger playing field

The lesson from IBM

- 1981 – the IBM 5150 (“Personal Computer”)
- off-the-shelf parts (“LEGO”) with open architecture, the BIOS was copied by others
- the IPR controversy – others started to produce “PC-compatibles” without paying revenues
- Others had no chance on market (even technically more advanced Macs)
- The king of personal computer landscape for over 20 years by now

Richard Matthew Stallman (aka RMS)

- *The last of true hackers* (Levy)
- 1979 breakup of the AI Lab – LMI vs Symbolics
- One-man war
- 1983 – GNU project
- 1985 – the GNU Manifesto
- 1989 – the initial version of GNU GPL

GNU General Public License (GPL)

- Development:
 - 1989 – initial version
 - 1991 – version 2
 - 2007 – version 3
- Based on author's rights, but stresses the user rights rather than proprietary conditions for use
- Used for about 70% of free software

The four basic rights

- the right to use, copy and distribute the work for any purpose (including business!)
- the right to study the work - for software, demands inclusion of the source code
- the right to modify the work and develop new works based on it
- the right to distribute the derived works under the same conditions

Copyleft

- “Copyleft: all rights reversed” - originally a typical hackerly pun
- one of the main ideas behind the free culture
- nowadays means the obligation to carry the license over to the derivatives
- Can be
 - strong – all derivative works share the same license (GPL)
 - weak – there can be exceptions (LGPL)
 - absent – derived products can be non-free (BSD)

In short

- GPL-ed software can be freely copied, changed and redistributed provided that
- the same rights are exercised on derivatives - therefore, it is not possible to use GPL-ed software to develop a proprietary product (and “soon you'll have to pay for Linux!” is nonsense)
- the source code must be available - usually it is distributed with the work, but the license also permits on-demand distribution. In this case, the software must include clear guidelines how to obtain the source (e.g. by e-mail or web)

Some points

- if a GPL-ed product is changed only for 'in-house' use, publishing the changes is not mandatory. However, as soon as the derivative will be distributed outside, the source must be made available
- GPL directly forbids any discrimination on use - this also means that all commercial use is fully legal. Both original and derived software can be sold (done by e.g. Red Hat).
- GPL has a clause about the work being distributed 'as is', i.e. having no warranty. Actually, the market for these services is completely open!

GPL v3

- Summer 2007
- After initial debates started to gain ground
- Improves compatibility with some other free licenses
- Includes anti-patenting clauses – essentially, any party who is using software patents to attack others, will void its GPL v3 licenses (loses the right to use any GPL v3 -licensed software)

LGPL

- Earlier “Library GPL”, now “Lesser GPL”
- Weak copyleft license – allows some mixing with proprietary software (linking)
- LGPL code can be reused under GPL
- Notable users: OpenOffice.org, Mozilla (using multi-licensing)

AGPL

- First version by Affero Inc, 2002, current version 3 by FSF 2007 (in parallel with GPL 3)
- Used to close the ASP (application service provider) loophole in GPL 2 – especially in web applications
- Defines using over the network as equal to downloading and running locally

Why don't the two Steves like GPL

- it does not let businesspeople to use free software authors as unpaid labour force for developing their proprietary applications (ct. BSD license)
- it can be legally enforced (even to the point that most violations are settled before the trial)
- it is resistant to the “Embrace, Extend and Extinguish”, a major method for some companies

Other free licenses

- While GPL is dominant, there are dozens of others:
 - BSD 2.0, Apache, X11, MIT (non-copyleft)
 - Mozilla PL, MS-PL, MS-RL (weak copyleft)
 - CeCILL, EUPL (strong copyleft)
- Two schools of thought – FSF vs OSI – have also got their own license recognition lists

European licenses

- GPL and others are written with US legislation in mind
- In recent years, a couple of European copyleft licenses have been introduced:
 - CeCILL 2.0 – France 2005, compatible with both GPL 2 and 3 as well as French legislation, official versions in both French and English. There are also CeCILL-B (non-copyleft) and CeCILL-C (weak copyleft)
 - EUPL – EU 2009 (latest), compatible with GPL 2, allows relicensing under 5 well-known licenses (including CeCILL – indirect compatibility with GPL 3). All 22 translations are validated (incl Estonian)

Whose freedom is more important?

- GPL and other strong copyleft licenses – somewhat limit the freedom of developers in order to guarantee the freedom for users
- BSD, MIT and other non-copyleft licenses minimise the limitations for developers but may put the users' freedom at risk

A word of caution

- ...especially for developers
- “It's all free software, so we can mix it, right?”
- NOT ALWAYS
- The main criteria is the usage and level of copyleft. For example:
 - One module is under GPL – all derivatives must be GPLed
 - Another is under BSD 2.0 – does not stick
 - The result will probably go under GPL
- The best bet is to use components with same or similar licenses

Conclusion

- Free licenses are not about 'do what you want, no one cares' – rather they are about maximising the distribution (and market) as well as guaranteeing user rights
- Free licensing:
 - Old ethic, new technology
 - Different, but legally enforceable
 - Base for the emerging free culture

Additional reading

- Free Culture by Larry Lessig
- Information Liberation by Brian Martin
- Anarchism Triumphant by Eben Moglen
- WIPO.int website (traditional model)
- FSF and OSI license lists

Homework

- Study the principle of copyleft and write an analysis about its three variants (strong, weak, none) with real-life examples (different cases).
Due: Monday, October 3



</rant>
