The Use of Open Source Software Licensing in Academia

Stephen J Marshall

What is Open Source Software?

- Computer software in which the source code (the list of instructions in human-readable form) is freely available and included as part of the software distribution.
- OSS licenses permits others to view, use, modify and redistribute the source code.
- May also include a ‘copyleft’ clause, whereby all modifications and extensions to the original code must also be made available under the same terms.
- Not the same as placing it in the public domain.

University of Glasgow

Origins of Open Source Software

- Harks back to a time before the software industry existed.
- Origins lie in the ‘hacker’ culture of US computer science labs in the 1970s and the free exchange of software based on the UNIX operating system which took place during that period.
- Led to the formalisation of OSS principles of openness and cooperation and the establishment of organisations to support and promote OSS, beginning with the Free Software Foundation (FSF) in 1985.

The OSS Movement

- Two camps:
  - Those who believe that software should be open in order to improve quality and interoperability (“*Given enough eyeballs, all bugs are shallow*”). For them, OSS is primarily a development methodology. This view is represented by the Open Source Initiative (OSI).
  - Those who believe that software should be free, both in monetary terms and in terms of the freedom to do anything they like with it. For them, OSS is an ideology. This view is represented by the FSF. Recently extended to other types of creative works in digital form, such as music and video files that are protected by DRM technology.

Why is OSS Attractive?

Attraction for Developers

- A way of harnessing the resources of a greatly expanded development team when creating new code, leading to significant improvements in the speed of development.
- A rich source of ready-made software routines that they can build into their own projects and new ideas that they can learn from.
- Makes it easier for them to see how other software works, thereby improving interoperability and maintaining backward compatibility.

Attraction for End Users

- No license fees!
- For organisations whose reliance on a particular software product may be critical to their business, the risk of inadequate or diminishing support for the product (e.g. through obsolescence or if the vendor goes bust) is minimised.
- Option of supporting the software in-house, which also provides flexibility to customise it or port the code to new hardware if necessary.

Issues Surrounding OSS

Commercialisation Issues

- An OSS license does not prevent anyone from selling the software but no one will buy it if the source code is freely available, unless the seller is also providing some kind of added value.
- No exclusivity, as the seller cannot prevent anyone else from selling the software too.
- Standard business model is to sell support, documentation or consultancy services. This works for Linux but is not viable for niche market software.

Intellectual Property Rights – 1

- Idea of a community of individual developers working together on OSS projects is based on the assumption that each is free to contribute their IP to the project.
- Seldom true, as majority of contributors are likely to be employees of one kind or another and IP generated by an employee through the course of his or her employment legally belongs to the employer.
- Some will argue that IP generated at home or without the use of their employer’s resources is exempt.

Intellectual Property Rights – 2

- This argument might be successful where an employee’s duties don’t include software development but the employee should obtain the prior written agreement of their employer in order to avoid the possibility of a legal dispute.
- Lack of awareness of this issue within the OSS community and the absence of any due diligence when releasing code under an OSS license suggests that a significant proportion of OSS projects may be tainted by IPR problems.

University of Glasgow

Professionalism

- If software professionals are unable to contribute their IP to OSS projects, who can?
  - Academics employed by universities are also unlikely to own their IP (although their employers may be more willing to allow them to participate).
  - Self-employed and contract software engineers aren’t usually bound by employer’s IP rights but are unlikely to be motivated to write OSS unless they can earn a living from it, and the unpaid volunteer nature of OSS tends to rule this out.
  - Students and self-taught amateurs are free (and positively encouraged) to contribute.
- Movement doesn‘t promote professionalism.

Conceptual Integrity

- Software needs a lead designer (or software architect) with a clear design concept which must be adhered to rigorously otherwise code will lose structure and become more difficult to manage as it is developed in a piecemeal manner.
- Peer review is, or should be, part of the conventional software engineering process and is no substitute for a coherent architectural vision which the community-based model of software development does not foster.

Innovation

- Noticeable lack of imagination in OSS projects, perhaps explained by the absence of design leadership in OSS development and the temptation for OSS developers to create free versions of their favourite proprietary software.
- Linux is an excellent case in point. Despite containing many powerful new tools and utilities, it is essentially a facsimile of UNIX, the proprietary operating system originally developed at Bell Labs in 1969.

Industry Concerns

- A continued shift towards OSS solutions at the expense of proprietary ones could result in many of the companies that develop proprietary software going out of business.
- The software industry doesn’t seem to have recognised this and is currently using OSS in an attempt to destroy Microsoft’s market dominance.
- The first companies affected are likely to be the small but highly innovative firms that are the lifeblood of the software industry, not the giant corporations that we all love to hate.

Use of OSS in Academia

Institutional Drivers – 1

- In 2004, the Office of the e-Envoy and the DTI recommended the adoption of OSS as the default exploitation route for UK government-funded R&D software outputs.
- If no commercial or community-shared exploitation route is specified in the final report, an OSS default will apply for all software developed under any UK government-funded research project.
- Could lead to situations where software remains undeclared until after the end of the project, whereupon the default exploitation route automatically comes into effect.

Institutional Drivers – 2

- Argument that all university-originated software should be released as OSS to maximise the benefit to industry but software licensing can be a significant source of revenue for universities (Stanford has earned $336 million from licensing search engine technology to Google).
- To minimise the risk of research software with significant commercial potential slipping through the net, it is important for universities to have a clear institutional policy on OSS.

A Mechanism for Knowledge Transfer

- Releasing university-originated software under an OSS license can:
  - Stimulate new research collaborations.
  - Free up source code for teaching purposes and allow it to be included in research papers and textbooks.
  - Protect the long-term future of critical research software by making it more widely available, thus increasing the number of users and developers.
- Also an efficient mechanism for KT, as it requires less effort than licensing software on commercial terms and is a faster, more direct route to market.

Academic Motivation

- OSS is attractive to many academics whose research involves software development.
- Ideologically motivated academics may be less inclined to consider alternative licensing options and may insist that all the software that they create should be released as OSS.
- In such cases, it may be necessary to remind them that they have no authority to issue licenses on behalf of the institution and that they should work with their TTO to choose the most appropriate license to suit the circumstances.

Choice of OSS License – 1

- UK government policy states that only OSS licenses which are compliant with the Open Source Definition can be used. This is managed by the Open Source Initiative.
- Obtaining OSI approval for a university license would be proof of compliance but considerable time and effort would be required to draft the new license.
- A simpler option is to use an existing license that has already obtained OSI approval.

Choice of OSS License – 2

- The OSI web site currently lists 66 OSI-approved licenses. These can be grouped into two basic types;
  - Licenses that permit others to view, use, modify and redistribute source code but place few additional restrictions on developers other than that they retain the original copyright notice. Best known example is the BSD license.
  - Licenses that include a copyleft clause, thereby constraining developers from making the code proprietary or combining it with other code that is licensed under different terms. Best known example is the GPL.
- Also a third type, which is a compromise between both in order to permit OSS and proprietary code to be mixed under certain conditions. Best known example is the MPL.

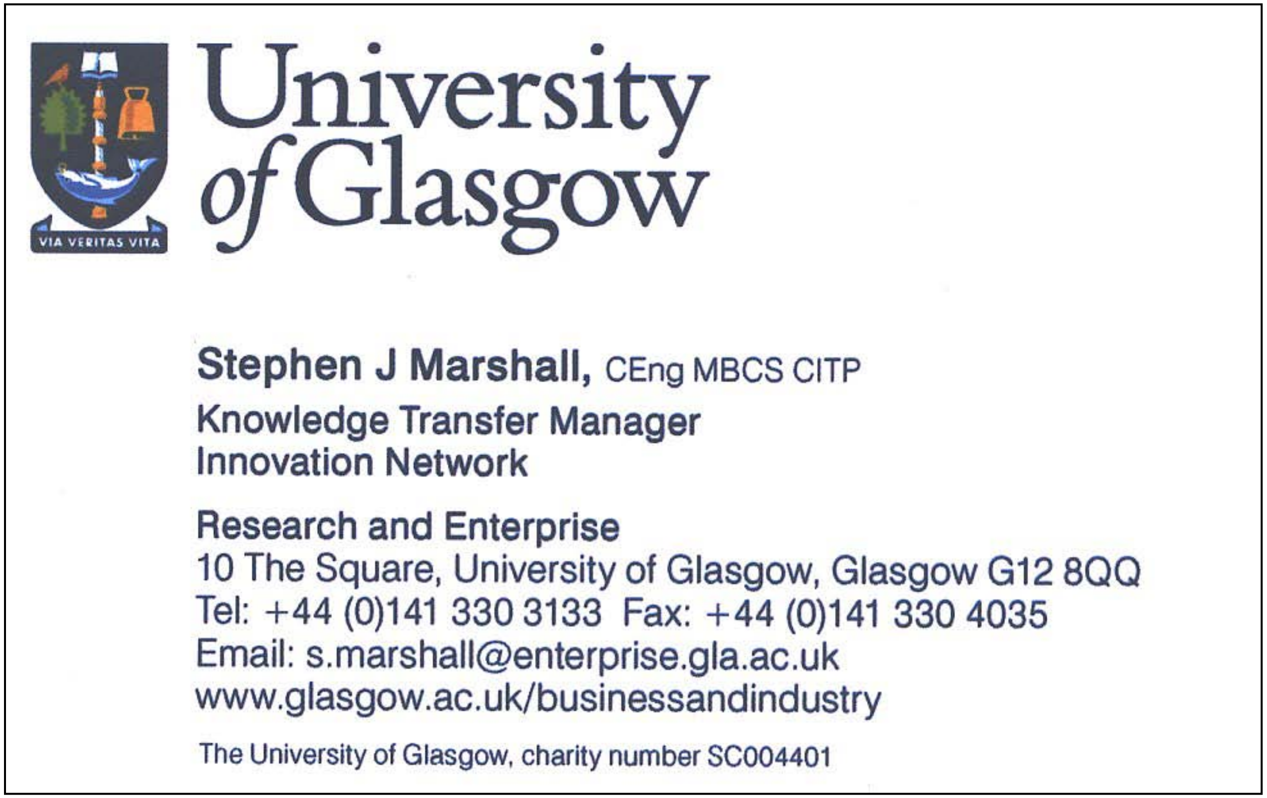
Choice of OSS License – Typical Scenarios

1. Software has no commercial value but the originators wish to encourage other groups to use it:
   - Any of the three types of license would suit but the MPL is probably best as it retains the copyleft provisions necessary to allow the originators to benefit from further development of the code by other groups but is less extreme than the GPL.
2. Software has commercial potential but there are no prospective licensees on the horizon and it would be difficult to justify withholding an OSS release:
   - The MPL would also be an appropriate choice as it doesn't restrict the use of the software in the development of a proprietary product but still retains the copyleft provisions.
3. Specific requirement to prevent competitors from creating proprietary versions of university-originated software released as OSS:
   - The GPL would be the logical choice but it should be used with extreme caution as it would effectively prevent any commercialisation whatsoever of the software or its derivatives under any circumstances.

Conclusions

- Releasing university-originated software under an OSS license can be an effective mechanism for KT when used appropriately.
- However, there are pitfalls and its use should be carefully managed.
- Key requirement is an institutional policy that is sympathetic to the needs of academics who wish to engage in OSS development but doesn’t restrict the institution’s ability to generate licensing income in cases where there is significant commercial potential.

Contact Details



University of Glasgow

Stephen J Marshall, CEng MBCS CITP

Knowledge Transfer Manager
Innovation Network

Research and Enterprise
10 The Square, University of Glasgow, Glasgow G12 8QQ
Tel: +44 (0)141 330 3133 Fax: +44 (0)141 330 4035
Email: s.marshall@enterprise.gla.ac.uk
www.glasgow.ac.uk/businessandindustry

The University of Glasgow, charity number SC004401