An Open Source license for the CCP in Synergistic PET-MR Reconstruction

Author: Kris Thielemans (KT), contributions and suggestions by Andrew Reader

Disclaimer: the license descriptions are based on KT's current understanding but have not been reviewed by a lawyer.

Motivation

One of the main outcomes of the CCP will be the software that we integrate and develop. We want to **encourage collaboration and usage** while also providing a **clear and safe legal framework**. To achieve this, we need to

- make licensing and all associated processes as easy as possible (*e.g.* a single license is preferred).
- use an existing and well-known license (this is clearer for the user, such licenses have been designed by people with lots of legal resources and there is "safety in numbers" when it would get to litigation).
- impose minimal restrictions on the users of the software (including ourselves).

In this document, we give a brief introduction to open source software licenses and provide our motivation for recommending the Apache 2.0 license.

Mini-tutorial on copyright and licensing

Whenever you write software (or a book), someone automatically owns the copyright for that work. This owner can be you, but in the UK it is normally your employer (students can own the copyright for what they write, but often are asked to assign it to the university).

Copyright means that nobody else is allowed either copy or use that software unless the copyright holder provides them with a license. A license gives certain rights to the user (e.g. "*you can use the software for such-and-such purpose*", or "*you can distribute the software only as-is*", or "*you can distribute your own modifications*"), but also protects the copyright owner by imposing conditions (e.g. "*you can only distribute your modifications if you preserve my copyright notice*", or "*I provide no warranty whatsoever*" or "*if you use this software, you cannot sue me for breaching one of your patents with this software*").

Note that the copyright owner can license the software to different people with different rights/conditions (e.g. a "free" license to academics, a "you have to pay me" license to a company).

If multiple people from multiple institutions work together on the same software, some agreement on licensing is necessary, otherwise they are (in principle) not allowed to use each other's software.

Open Source Licenses

There exist a very large number of open source licenses. Most, possibly all, modern open source licenses allow the user to *use* the software for any purpose and have access to the original source code, while disclaiming warranty and providing patent protection. The differences between the licenses are mostly related to rights for *redistribution*. Open source licenses do not impose restrictions like "not for commercial use" (but they might discourage it by imposing conditions on redistribution). Modern licenses also no longer enforce attribution (except in the source code and usage messages etc), as this soon becomes unwieldy. For instance, which paper would you refer to if you use ITK-based Slicer3D as its methods have been described in many different papers?

Here we describe only 3 licenses currently used for related work[1]: GPL, LPGL and Apache 2.0.

Comparison of the GPL, LPGL and Apache 2.0 licenses

GPL and LGPL

See http://www.gnu.org/philosophy/license-list.html#GPLCompatibleLicenses

These licenses were developed by the Free Software Foundation http://www.fsf.org/, a non-commercial organisation that has a long track-record in providing and advocating open source software. Both GPL and LGPL require that any redistributed

[1] NifTk (UCL), http://cmic.cs.ucl.ac.uk/home/software/ and ITK http://itk.org/ use the Apache 2.0 license. KCL's image registration software used for their PET-MR work is "2 clause BSD" (which is less restrictive than Apache 2.0). STIR http://stir.sourceforge.net uses the GNU public license (GPL) for applications, and the Lesser GPL (LGPL) for the library part. However, STIR will in the near future move to Apache 2.0 as well (after agreement of all copyright holders). CCPi will use LGPL for most software but is willing to consider using Apache 2.0 for components of interest to us (depending on commercial viability). The Gadgetron (MRI reconstruction software) uses an NIH license without any restrictions.

derivative of the licensed software has to have a license compatible with the original one (in particular, enforcing making source code available).

The main difference between the GPL and the LGPL is that the LGPL allows “closed” (in fact, non-LGPL) programs/applications to use/link with the LGPL software while the GPL forces any application using the software to be GPL'ed as well. The GPL is therefore sometimes called a “viral license” as it was designed to spread the adoption of open source practices.

Apache 2.0 license

See http://www.apache.org/licenses/LICENSE-2.0.html
This license is a “BSD-style” license. It imposes less conditions than the LGPL, in particular allows unrestricted use, including use in commercial/closed products. It does require that copyright notices are preserved, while warranty is disclaimed. It’s designed to be easy to apply in practice.

Table 1: brief overview of licenses, ordered from less to more restrictive

License	Conditions on redistribution	Remarks
BSD	Preserve copyright and attribution notices, do not use name of copyright owner for promotion of derived products, no warranty.	Allows “closed source” redistribution (e.g. as part of a commercial package)
Apache 2.0	Above conditions + patent protection + mark files that the user changed. Explicitly allows providing warranty for a fee.	More modern license with more protection. Allows “closed source” redistribution (e.g. as part of a commercial package).
LGPL	Above conditions + distribute source code + use LGPL for the modifications.	Allows “closed source” applications to use the software, but the original “open source” part needs to remains so. Allows
GPL	Above conditions + use GPL for whole application.	Whole application needs to be “open source”, even entirely new code

Pros and cons

GPL:

- strongest conditions on redistribution while being open source (but therefore also more restrictions on using incorporated code from others)
- makes it possible to use other GPL libraries
- can only be used in GPL applications
- prevents use of “non-GPL compatible” software, and in particular closed software. This would exclude using software from a manufacturer (or a MATLAB toolbox) as part of the distributed “application”.
- vendors usually do not want to use GPL software as it prevents them to keep certain parts confidential and (almost?) invalidates patent protection.

LPGL:

- makes it easier than GPL to incorporate external contributions as there are fewer restrictions
- allows use in LGPL and GPL applications
- an application that uses the LGPL software can also use closed/commercial components or could itself be commercial

Apache 2.0 license:

- nearly complete freedom for everyone, therefore the best way to attract interest from commercial companies
- same type of license as ITK and NifTK and KCL registration tools
- allows use in LGPL and GPL applications
- easy to apply (less editing of headers etc)

Proposal

Based on the previous arguments, we recommend using the Apache 2.0 license.