Published in:
Möller, A.; Page, B.; Schreiber, M. (2008): 22nd International Conference on Informatics for Environmental Protection, Environmental Informatics and Industrial Ecology, Shaker Verlag, Aachen.

Odyssey to Kalypso – Experiences with setting up an Open Source Platform for Environmental Modeling

Gernot Belger[1], Michael Haase[1], Kaj Lippert[1]

Abstract

In recent years a large number of open source solutions emerged which are of special interest to Environmental Informatics. Open source is now no longer just a playground of universities, research institutes and committed individuals who are communicating via the Internet to make ideas come to reality. Open source is also increasingly reflected in the business models of small and medium seized enterprises as well as well-known large IT companies.

Different viewpoints on the issue of open source software from the Open Source Initiative and the Free Software Foundation are presented and analyzed. An introduction to open source license categories is given and compatibility issues of popular and widely used open source software licenses are discussed. The paper also introduces to basic strategies how to set up an open source project.

A certain charm lies in the creation and establishment of one's own open source project. This paper presents the story (Odyssey) of the open source project Kalypso (http://kalypso.sourceforge.net/). Kalypso encompasses a software palette for water resources modeling (e.g. hydrologic and hydraulic models) which was built on the basis of the powerful Kalypso framework. This framework allows to build extensive decision support and information systems including among others spatial data management and analysis, time series management and analysis, generation and management of reports, database access and accessing existing executable numerical models from the fields of environmental and water resources modeling. The paper introduces to the Kalypso project and presents the work spent to publish Kalypso on an open source platform.

1. Introduction – Background

Open source software is increasingly being used by a wide range of applicants from Universities, research institutes and IT firms as well as company's IT departments for building specific applications. Ready-to-use open software applications are provided by an increasing number of suppliers from industry, Universities and research institutes. Increasingly employers from industry as well as from public authorities request that customer tailored applications shall either be built on the basis of opens source components and shall be made available to other users on the basis of open source agreements respectively. A multitude of literature is available covering the topic ranging from using open source software for the enterprise (e.g. Woods & Guliani 2005) to producing and managing open software projects successfully (e.g. Sandred 2001, Fogel 2006). There are various definitions of open source software about which are controversial in nature, e.g. Free Software Foundation (FSF 2008) and Open Source Initiative (OSI 2008).

[1] Björnsen Consulting Engineers GmbH, Maria Trost 3, D-56070 Koblenz, Germany,
email: {g.belger | m.haase | k.lippert}@bjoernsen.de

The OSI lists ten criteria which ought to be met in order to be “open source software” (OSI 2008):

(1) Free redistribution – Software components from different sources may be used to build a new peace of software and distribute this software with no royalties or fees to worry about.

(2) Source code – The source code as well as a compiled form of the new piece of software must be available either for free or some reasonable reproduction cost.

(3) Derived works – The original code may be modified and distributed taking the license of the original software into account.

(4) Integrity of the author's source code - The license of the original code must explicitly permit distribution of software built from modified source code and other requisites of this license need to be taken into account.

(5) No discrimination against persons or groups - The license must not discriminate against any person or group of persons.

(6) No discrimination against fields of endeavor - The license must not restrict anyone from making use of the program in a specific field of endeavor.

(7) Distribution of license - The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

(8) License must not be specific to a product - The rights attached to the program must not depend on the program's being part of a particular software distribution.

(9) License must not restrict other software - The license must not place restrictions on other software that is distributed along with the licensed software.

(10) License must be technology-neutral - No provision of the license may be predicated on any individual technology or style of interface.

According to these criteria, “open source software” is more than just being able to access the codes of a piece of software. This is more than just a philosophical question. In particular, when the IT industry releases open source software there is a business model behind these activities. Open source software in this context is different from a basic right, e.g. “freedom of speech”. There are restrictions about for deploying “open source software”. On the other hand Richard Stallman – a founder of the GNU Project (GNU 2008) and the Free Software Foundation (FSF 2008) - refers to “free software”. “Free software” in this context meets the following criteria (FSF 2008):

(1) The freedom to run the program, for any purpose (freedom 0).

(2) The freedom to study how the program works, and adapt it to your needs (freedom 1).

(3) The freedom to redistribute copies so you can help your neighbor (freedom 2).

(4) The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).

Access to the source code is a precondition to meet “freedom 1” and “freedom 3”. The points of view of looking from the OSI and the FSF at the issue of “open source” are quite different. In a matter of fact, these definitions are not compatible at all in some cases, e.g. criterion number 4 of the OSI contradicts with criterion number 3 of the FSF. The authors agree with the criteria of “open source software” from the OSI; although in the process of building “open source software” these criteria can impose a burden on decision makers as well as software developers.

This paper is set up as follows: After the excurse to different viewpoints on the issue of “open source” different categories of open source licenses and interoperability issues of licenses are discussed in chapter 2. Chapter 3 introduces to general requisites for setting up an open source project. The open source project

Kalypso and experiences with setting up Kalypso as an open source project will be introduced in chapter 4. In Chapter 5 a brief summary of this paper including a perspective for Kalypso is compiled.

2. Open Source Licenses

Chapter 1 already introduced the "license" issue and different viewpoints of looking at "open source". In order to illustrate this point some popular open source projects and their licenses are presented hereafter. Big IT companies have contributed a multitude of open source projects to the open source community within the last decade. Some have worked out special licenses to go along with the source codes of their products. Popular examples of this practice are: Netscape has given away its source code to the Mozilla Organisation under the Mozilla Public Licence (Mozilla 2008) in 1998, Sun Microsystems started the open source OpenOffice.org project in 2000 and donated their office software suite StarOffice to the project which is now licensed under the GNU Lesser General Public License (OpenOffice.org 2008), IBM released the Eclipse Platform as open source and launched with other partners from the IT industry the Eclipse Foundation which now provides technology and source codes under the Eclipse Public License (Eclipse 2008). Other very successful open source projects started in a grass root type of way: In 1999 the Apache Software Foundation was founded to provide support for the Apache HTTP Server which is provided under the Apache License (Apache 2008) - a great number of other open source software projects now fall under the Foundation's umbrella as well - , the development of Linux is another prominent example in this category which is provided under the GNU General Public License (Linux 2008).

The Open Source Initiative lists about 70 different open source licenses and classifies these licenses into the categories listed hereafter (OSI 2008):

- Licenses that are popular and widely used or with strong communities - Licenses in this group were selected on the basis of statistical analysis of public sources.
- Special purpose licenses - Licenses that were identified as meeting a special need were placed in this group.
- Licenses that are redundant with more popular licenses - Licenses that were perceived as completely or partially redundant with existing licenses are placed in this group.
- Non-reusable licenses - Licenses in this group are specific to their authors and cannot be reused by others.
- Superseded licenses - Licenses in this category have been superseded by newer versions.
- Licenses that have been voluntarily retired – Licenses that should be used any more in the future.
- Other / miscellaneous licenses - These licenses do not fall into any category.

In the context of this paper we are just interested in the first category which contains nine different licenses (OSI 2008). These licenses including their founder are listed hereafter:

- Apache License 2.0, Apache Foundation
- BSD License, University of California, Berkeley
- Common Development and Distribution License (CDDL), Sun Microsystems
- Common Public License 1.0 (CPL), IBM
- Eclipse Public License (EPL), Eclipse Foundation
- GNU General Public License (GPL), Free Software Foundation
- GNU Library or Lesser General Public License (LGPL), Free Software Foundation
- MIT License, Massachusetts Institute of Technology
- Mozilla Public License 1.1 (MPL), Mozilla Foundation

A survey from the authors about licenses being used within open source projects listed at the SourceForge.net (SourceForge 2008) carried out in early June 2008 shows that from an arbitrary sample of 125.342 open source projects at that time only 15.353 (about 12 %) use other licenses than those contained in this list. Figure 1 shows a bar chart with the results of this survey. The ordinate is displayed in a logarithmic scale due to the large number of GPL licensed projects. The figure clearly shows that the most frequently used licenses in projects are GPL and LGPL followed by BSD, Apache and MIT licenses. SourceForge.net allows for a multiple allocation of licenses per project.

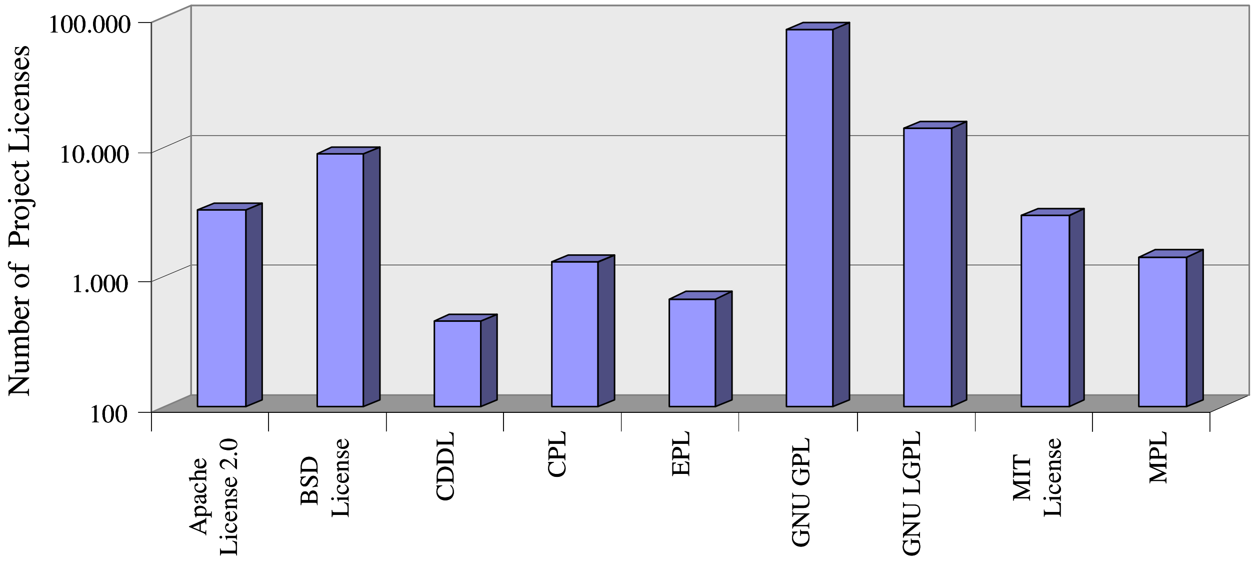

Fig. 1: Number of referenced popular and widely used licenses at SourceForge.net (survey from the authors from early June 2008)

Tab. 1: Compatibility of popular and widely used licenses (modified after Rusin 2008)

License	Proprietary software linking	Distribution with one's own work	Redistribution of the code with changes	Compatibility with GNU GPL
Apache License 2.0	allowed	allowed	allowed	no
BSD License	allowed	allowed	allowed	yes
CDDL	allowed	allowed (source codes need to be provided)	only under CDDL	no
CPL	allowed	allowed	only under CPL	no
EPL	allowed	allowed	allowed	no
GPL	not allowed	not allowed with software whose license is not GNU GPL compatible	only if the derivative is GNU GPL	yes
LGPL	allowed	allowed (source codes need to be provided)	only if the derivative is GNU LGPL or GNU GPL.	yes
MIT License	allowed	allowed	allowed	yes
MPL	allowed	allowed	only under MPL	no

Using open source software requires a sound understanding of the licenses attached to these pieces of software. In this context the understanding of the term "copyleft" is of special importance (Wikipedia 2008): "Copyleft is a play on the word copyright and describes the practice of using copyright law to remove restrictions on distributing copies and modified versions of a work for others and requiring that the same freedoms be preserved in modified versions." "Strong copyleft" and "weak copyleft" are differentiated (Wikipedia 2008): "The copyleft governing a work is considered to be "stronger", to the extent that the copyleft provisions can be efficiently imposed on all kinds of derived works. "Weak copyleft" refers to licenses where not all derived works inherit the copyleft license; whether a derived work inherits or not often depends on the manner in which it was derived." The GNU General Public License uses "strong copyleft". "Weak copyleft" include the GNU Lesser General Public License and the Mozilla Public License. Examples of "non-copyleft" software licenses include the MIT, Apache, Eclipse and the BSD licenses. Table 1 provides an overview on license compatibilities taking into account issues of linking closed sourced applications with applications/libraries licensed (Proprietary software linking), combining software with licensed libraries/applications (Distribution with one's own work) and redistribute modified applications/libraries under the given license (Redistribution of the code with changes). The last column lists the compatibility with the GNU GPL license. Table 1 clearly shows that setting up a software project on the basis of open source software needs special attention from the management in order to account for the compatibility of different licenses being used in that particular software project.

3. Setting up an Open Source Project

Setting up an open source project should be planned carefully. Within the framework of this chapter only a short introduction into the topic of setting up an open source project can be provided, for a detailed discussion see e.g. Sandred (Sandred 2001) or Fogel (Fogel 2006).

First of all the question needs to be answered whether a new project shall be started or whether one should rather contribute to an existing project. Another issue which needs to be decided on is whether other parties may participate in the development process or whether they are not entitled to do so. Projects of the latter type are of course easier to handle, but this approach very probably limits the innovation and development process of the project which is vital for an open source project.

Marketing aspects need to be considered when setting up an open source project: The project needs a unique, easy to remember name and a good website to inform about the project. The website shall among others provide information on the goals of the project, planned future developments in the project and list prominent examples of the application of the project results. It shall also provide means to get into contact with the project management.

A software license (see previous chapter) for publishing the code needs to be chosen. Code versioning mechanisms must be set up in order to manage code updates.

The source codes as well as code manuals and sample data need to be published. Managing bugs which have been reported is a vital issue in every software project. In the context of open source projects, the bug fixing rate can be interpreted as an indicator of project progression. Mailing lists for informing users and developers about active developments in the project are needed. And last not least thematic discussion groups may help to share knowledge about project related topics within the project user group.

There are basically two different ways for implementing such a portal: An internet portal with all these features can be set up on the basis of e.g. open source software and hosted for public access on a private server or the project can be hosted on a publicly accessible open source portal which includes all of the afore mentioned features and is free of charge. Prominent examples for the latter are the SourceForge.net (SourceForge 2008) and freshmeat (freshmeat 2008) open source portals.

4. Kalypso

4.1 Introduction to Kalypso

The Kalypso Project is based upon Kalypso Base which forms a framework for building Java-based "rich client" decision support and information systems. On this foundation a series of Kalypso applications for water resources modeling have been developed which are also available as open source from the Kalypso Project:

- Kalypso Hydrology – A distributed rainfall-runoff model for simulating hydrological processes within a catchment.
- Kalypso WSPM – A one-dimensional backwater surface profiling model for one-dimensional non-uniform steady fluvial flow.
- Kalypso 1D/2D – A hybrid 1- and 2-dimensional Finite Element model for non-steady flow simulation in rivers and estuaries.
- Kalypso Flood – A tool for visualizing flooded areas determined either by Kalypso WSPM or Kalypso 1D/2D.
- Kalypso Risk – A tool for determining the expected annual damage and classifying flooded areas into risk zones.

Kalypso Hydrology, Kalypso WSPM and Kalypso 1D/2D are each comprised of three parts: a data pre-processing component, a numerical processing component – the numerical processing software is available as freeware with Kalypso - and a post-processing component for presenting and analyzing data obtained with the numerical processing component.

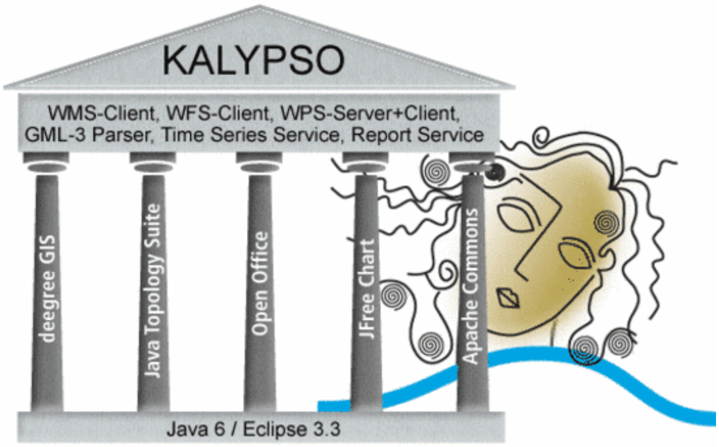

Fig. 2: Kalypso Base

Kalypso Base resembles a Greek temple (see figure 2). Its foundation is based on the Java programming language which includes the corresponding Java virtual machine and the Eclipse (Eclipse 2008) development environment. The pillars in this analog among others are the deegree GIS (deegree 2008), the Java Topology Suite (JTS 2008) for processing geodata, the Open Office package (OpenOffice 2008), the JFreeChart (JFreeChart 2008) collection and the Apache Commons library (Apache Commons 2008). A complete list of third party libraries can be found at http://kalypso.sourceforge.net/legal/libraries.html. On top rest the architrave and frieze, the latter carries the facets OGC WMS-Client, OGC WFS-Client, OGC WPS-Server and OGC WPS-Client, GML-3-Formats (for details about these services and formats please refer to OGC 2008). Kalypso Base encapsulates these features, it represents the tympanum within the cornice of the temple.

Extensive other decision support systems were built on the foundation of the Kalypso Base, e.g. INFORM (Hens & Rosenzweig 2008), nofdp IDSS (Löw et al. 2008). The latter is available as open source as well (nofdp IDSS 2008).

4.2 Experiences with setting up the Kalypso Open Source Project

Kalypso evolved from a client project a few years ago and was elaborated continuously ever since into an open source software project with powerful features. After initially deciding on the name Kalypso and building a temporary website as well as a code download site it took us a while to decide whether to invest more time and money to build a professional open source project Kalypso.

First, we redesigned the existing Kalypso website. After this had been accomplished, it was decided to go on and publish Kalypso as an open source project. In this context we had to decide where to publish the project and how to license Kalypso: Since the software is based on Eclipse we were offered to publish it on the Eclipse platform in terms of the EPL. In order to do so we would have had to split up the codes which comprise Kalypso. In consequence, Kalypso users would then have been urged to compile the other programs/sources of Kalypso on their own which is an error-prone process and might have frustrated them. We did not want this to happen and therefore decided to publish the Kalypso project on the SourceForge.net portal in terms of the GNU LGPL. The reason for choosing the GNU LGPL license for Kalypso was that many of the open source libraries/software pieces used in Kalypso are licensed with the GNU LGPL; other components like Apache Commons and Eclipse are compatible with the GNU LGPL.

We had to invest quite a bit of work for publishing Kalypso as an open source project:

- All of our own source codes needed to be licensed.
- Third party libraries needed to be separated from our own codes.
- A deployment procedure was introduced which required a consistent handling of version numbers, release notes, change logs and source code branching.
- In order to improve the quality of the final product a distinction between release candidates and stable releases was introduced to the deployment process.
- Sample data needed to be compiled in order to publish them on the SourgeForge.net portal with the codes.
- A Kalypso project had to be defined and customized on the SourceForge.net portal (e.g. definition of user rights for accessing Kalypso codes and documentations for different user groups, upload the Kalypso source codes into SourceForge.net portal, initialization of a bug tracking system and different emailing services).
- Last not least, new internal processes had to be defined and implemented on the side of the developers (developer guidelines), e.g. formatting of codes, adding license texts to the codes, versioning rules, deploying new releases, code documentation, code testing procedures and documentation of the tests.

5. Summary and future Developments in the Kalypso Project

Setting up an open source project from scratch is a quite laborious task which needs careful management. In particular, the licensing issue has to be handled with special care in order to avoid incompatibility problems with licenses from third party software which is used in the project. New processes of building software need to be introduced to the software development teams and implemented on the long run.

Kalypso has been designed to meet the requirements which result from environmental and water resources modeling tasks. These requirements are quite extensive. Therefore, the future growth of Kalypso will be closely interrelated with the requirements of our customers. We like to invite other developer teams to share Kalypso with us and contribute new developments to Kalypso.

6. References

Apache (2008): http://httpd.apache.org/

Apache Commons (2008): http://commons.apache.org/

deegree (2008): http://www.deegree.org/

Eclipse (2008): http://www.eclipse.org/

Fogel, K. (2006): “Producing open source software – How to run a successful software project”, O’Reilly, Sebastopol, USA.

Free Software Foundation (2008): http://www.fsf.org/

freshmeat (2008): http://freshmeat.net/

GNU (2008): http://www.gnu.org/philosophy/open-source-misses-the-point.html

Hens, T.; Rosenzweig, S. (2008): “INFORM.DSS - Ein entscheidungsunterstützendes Informationssystem für die Wasser- und Schifffahrtsverwaltung des Bundes zur Berücksichtigung ökologischer Belange beim Gewässerausbau und der Gewässerunterhaltung“, in Gnauck, A. (Ed.): “Modellierung und Simulation von Ökosystemen, Workshop Kölpinsee 2007“, Shaker Verlag, Aachen.

JFreeChart (2008): http://www.jfree.org/jfreechart/

JTS (2008): http://www.vividsolutions.com/jts/jtshome.htm

Kalypso (2008): http://kalypso.sourceforge.net/

Linux (2008): http://www.linux.org

LGPL (2008): http://www.gnu.org/licenses/lgpl.html

Löw, M.; Belger, G.; Haase, M.; Hens, T.; Kuch, D.; Lippert, K.; Thül, M.; Hübner, C.; Ostrowski, M.; Winterscheid, A.; Fuchs, E.; Horchler, P.; Hüsing, V.; Rosenzweig, S.; Slikker, J.; van Iersel, P. (2008): “nofdp IDSS – Ein Informations- und Entscheidungsunterstützungssystem zur Planung von Hochwasserschutzmaßnahmen im technisch-ökologisch-ökonomischen-raumplanerischen Kontext“, in Gnauck, A. (Ed.): “Modellierung und Simulation von Ökosystemen, Workshop Kölpinsee 2007“, Shaker Verlag, Aachen.

Mozilla (2008):http://www.mozilla.org/

nofdp IDSS (2008): http://130.83.196.154/nofdp_idss/forum/

OpenOffice (2008): http://www.openoffice.org/

Open Source Initiative (2008): http://www.opensource.org/

OGC (2008): http://www.opengeospatial.org/

Rusin, Zack (2008): Open Source Licenses, http://developer.kde.org/documentation/licensing/licenses_summary.html

Sandred, J. (2001): “Managing open source projects”, John Wiley & Sons, Inc., New York, USA.

SourceForge (2008): http://sourceforge.net/

Wikipedia (2008): http://en.wikipedia.org/wiki/Copyleft

Woods, D.; Guliani, G. (2005): “Open source for the enterprise – Managing risks, reaping rewards”, O’Reilly, Sebastopol, USA.