Open Source for Land Management and Cadastre

Authors:
Arnulf Christl (Metaspatial, Germany, http://metaspatial.net)
Markus Neteler, (Mundialis, Germany, http://mundialis.de)

Background

When it comes to computer software, the old saying “you get what you pay for” may no longer apply. After years of skepticism towards open source software, many of today’s open source solutions are as good, if not better than proprietary software solutions. The question is- why is it then that there are so few land administration systems making use of open source software technology? Lack of knowledge about the possibilities might be one of the reasons. After all, marketing has never been a priority for developers of open source software. Doubts about the security and available software support could further shy away cadastre agencies from making the switch.

Land administration and cadastral systems are playing a crucial macro-economic role in the collection, management, and dissemination of information about land ownership, use and value. Cadastral systems are documenting land tenure rights and are thus providing crucial economic, social and environmental benefits. Modern cadastral systems make extensive use of information technology (IT) supported by software systems. In developed countries, such systems have been established over the last 20-30 years and became powerful tools in operating cadastral systems. In developing and transitional countries, the need for efficient cadastral systems and the use of IT is as much a necessity as it is in developed countries, although there are substantial financial and operational constraints.

The motivation for FAO, World Bank and FIG to become active in the field of open-source software for cadastre and land registration comes from the observation that many systems and projects in developing countries struggle to provide appropriate and affordable services for tenure security. Reasons are related to governance but also to technological and financial shortcomings. Information technology plays a crucial role in operating cadastres and land registration systems. In developing countries, the on-going license costs of proprietary software often created serious constraints and have even stopped programmes.

The costs of proprietary software licenses have proved to be a constraint, but even more, the lack of capacity, models and support to develop software have stopped initiatives. Open-source software, which has become a credible alternative to proprietary software, provides a way forward.

Open source solutions are more flexible and adaptable to local conditions and languages than proprietary software. By using and improving open source software, cadastres can build local knowledge and contribute to the development of open source projects that can in turn benefit other cadastres world-wide.

Free and Open Source Software

This short introduction to aims to give an insight to the inner workings of and Open Source development and Free Software licensing.

Free Software Licensing

The word "free" in Free Software refers to a degree of freedom and should not be confused with free as in gratis or in free beer. To make things a bit more complicated most software that comes with a Free Software license is available completely gratis or at a very marginal cost of a few cents for the actual download process. But the emphasis stays on the freedom of the user which is why we need to emphasize this again and again. With a Free Software license you are free to:

- use the software anywhere and for any purpose
- take it apart, understand and improve it
- pass it on to anybody else in both the original or a modified version
- make money by using it for any purpose
- improve it in exchange for a monetary compensation or for any other reason
- provide all kinds of services around it including training, installation, maintenance, etc.

These levels of freedom make up a Free Software license. For a comprehensive list of approved Free Software licenses please refer to the Free Software Foundation at http://www.gnu.org or the Open Source Initiative at http://www.opensource.org.

Proprietary Software

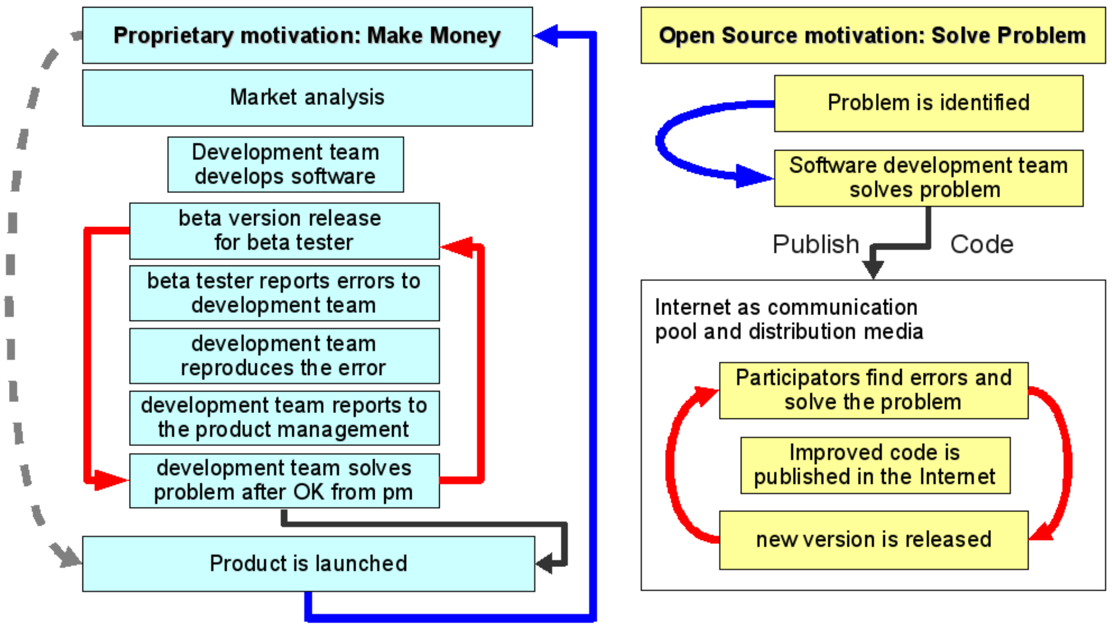
The opposite of **Free Software** is **proprietary software**. The single but very central difference between the two types of licenses is that in the latter case the proprietor (owner) of the software will restrict some or all of the above mentioned freedoms. You (the licensee) ~~is~~ are usually not allowed to use the software in more instances than is explicitly defined in the license contract. You are usually not allowed to take the software apart, or to modify it. You are not allowed to give the software away to anybody else. In some cases you are not allowed to make money by using the software in a certain way (by giving trainings or providing maintenance). In other cases you not allowed to provide services for the software without an additional license. Basically, proprietary licenses are designed to restrict freedom and explicitly take away the rights that are defined by the Free Software license model. This sounds bad (for you, the user) but it is actually not. It is just a very accepted but somewhat different business model and it has for some time been very efficient in generating revenue and even made one such proprietor the richest person in the world. But the proprietary software model is in the decline.

Open Source Development Model

In most cases "Open Source" can be used synonymously for "Free Software". For the sake of this introduction we will look at Open Source from a development model perspective. The source code of a software contains all the functionality in a human readable format. To change, enhance or extend the functionality of most software it is required that the source code be modified. Thus Open Source is a precondition to Free Software. End users will generally have no need to look into the source code and only work with the compiled, machine readable version. But it is still important to have the right to look into the software because only then can we fully understand what it is doing. Even if we do not, we can still pass it on to someone else who does have the capacity needed to understand the code. This will give you (the user) a degree of freedom from the monopoly of the vendor that proprietary licenses deny. All scientific research is based on absolutely transparent reproducibility which is not given if there is no possibility to look into the sources. Thus, strictly speaking, proprietary software cannot be used to analyze data for valid scientific research. Software developers naturally tend to drift to open development models because it makes reusing code a lot easier and allows for collaboration across organizational boundaries and between otherwise competing businesses. For many people these very basic facts are completely new concepts because they are not transparently communicated together with proprietary software.

Product and Development Cycle

The motivation to create and maintain Open Source software is inherently different to that of a product vendor (see: Illustration 1: Proprietary and Open Source Development Models). The left side of the illustration shows the typical development process of a vendor. The motivation of the vendor model is focused on making a profit. This will usually include a market study prior to starting the development. The development process itself is iterated in a closed environment until the software is released. The release date in most cases does not coincide with the software being ready to ship but with an event, for example a major industry trade show.
On the right side (Open Source) the intrinsic motivation is often to solve a problem at hand. If the problem is common then the resulting solution can be of use to others and over time a number of regular users (participants) may emerge. In this case the software is said to "take off" and it starts to get published on the web on a regular basis. New requirements appear as more users use the software in different contexts. The requirements may then be implemented in the order of need or availability of funding. If the project is successful, development will stabilize either through a growing user community or through one or more businesses that profit from continuing development on the software. The diagram shows some aspects of these differences.


Proprietary motivation: Make Money
Market analysis
Development team develops software
beta version release for beta tester
beta tester reports errors to development team
development team reproduces the error
development team reports to the product management
development team solves problem after OK from pm
Product is launched
Open Source motivation: Solve Problem
Problem is identified
Software development team solves problem
Publish
Code
Internet as communication pool and distribution media
Participators find errors and solve the problem
Improved code is published in the Internet
new version is released
Illustration 1: Proprietary and Open Source Development Models

Most noticeably the proprietary, closed development process limits the quality assurance to a restricted set of individuals. In the open development model everything is open to the scrutiny of many, which can result in the highest level of stability and security.
The obvious advantages of the Open Source development can be seen in the emergence and success of major projects like the Apache HTTP server (now running more than half of all websites globally). More specifically in the geospatial realm this effect can be seen in software packages like GDAL/OGR, PostGIS, Proj4, MapServer, GeoTools and many more. The open development model has so many advantages that all major proprietary vendors nowadays also naturally use the quick feedback mechanisms by asking users to fill out crash

reports. Results from these reports may then be distributed as patches through web technology which is exactly the way that it has been done in Open Source software development environments for many years. The difference here is again less transparency. While in an Open Source software project all current open and closed issues can be seen and analyzed and reacted to, proprietary vendors will usually keep them locked away.

Open Source and Security

At first sight being "Open" seems to contradict security because we are used to locking things away in the physical world to prevent them from being stolen. Thus proprietary code – that is itself essentially locked away – would also appear be more secure because none can look into its inner workings. But one of the very first paradigms of digital security says that security cannot be achieved by obscurity. Instead, all algorithms, architectures and concepts required to secure things must be open to the scrutiny of as many eyes as possible. This will ensure that they will get tested and verified by as many experts and in as many different settings as possible. Additionally it would not make any sense at all to try to lock away all protocols, code, software and architectures as "secret" because then no one can actually use and implement them. The only way out of this deadlock is to improve the software and architectures to the point that it is very, very hard to break them. This is exactly how security in the digital world works. All the encryption protocols that form the core security layer of the Internet are based on Open Source models. Real security has to stand up against being completely and thoroughly transparent. Only by opening up all process to the scrutiny of as many participants as possible a reliable system of security can be built.
Proprietary black box security systems might be hard to break to start with. But up to now in history every single security system ever has sooner or later been broken. Therefore the most important issue of digital security is to know **when** it has been compromised so that counter measures can be taken. For these reasons all major security systems that are nowadays in common use are based on Open Source models. This does not mean that proprietary software can by definition not be secure because it can implement the same Open Source algorithms, which is exactly what happens.
One example for geospatial Open Source adoption in a high security domain is the US Department of Defense (DoD). It was one of the major initial supporters of geospatial Open Source including the initial development of GRASS (the Geographic Resource and Analysis Support System). With the uptake of proprietary software in the 80s and 90s and the general need to reduce cost new business models emerged and the DoD turned to so called CotS (Commercial off the Shelf) software. The hope was to be able to reduce the TCO (total cost of ownership) by not developing software in-house but to rely on external sources. Recent studies conducted by the DoD evaluated the results of this strategy and show that the Open Source model is not inferior to the proprietary model from a financial perspective and that it is definitely superior with respect to security. As a result the DoD is again shifting its focus and has changed its documents [DoD, 2009] accordingly to make the use of Open Source in bids and tenders easier, stating that Open Source and proprietary software can synonymously be called "commercial software".

FLOSS Business Models

Usually FOSS Business Models are explained by listing activities that can be offered as a service. But it is a lot easier to work from the other end and acknowledge that all business models around software naturally apply to FLOSS except for exclusive proprietary licensing. Estimates show that less than 5% of all revenue generated by business activities around software is generated by selling proprietary software licenses [Bruce Perens, 2005]. On the other hand there are practically no reliable numbers that could quantify the positive net productivity effect of any given software as it is not possible to compare one over another in a reproducible environment.
One reason why Open Source models have been adopted early in the geospatial domain is the intrinsic interconnectedness of spatial data which relates well to the interconnectedness

of knowledge – and code is nothing but formalized knowledge. Especially in the geospatial domain a healthy business ecology has emerged as can be seen in the Servcie Provider Directory of OSGeo where a total of more than 150 companies are registered. This register represents only a fraction of all businesses that offer service, support, training, consultancy and maintenance for the whole range of spatially enabled software, ranging from the single contractor business to divisions of large enterprises that employ several hundred specialists.

FLOSS Adoption by the Industry

Unquestionably Open Source is the superior development model. This has been proven by all major software enterprises, one of its pioneers being IBM which recognized the emerging paradigm shift at a very early stage. Nowadays all major software vendors including Oracle and even Microsoft have at one point either purchased Open Source companies or product names or adopted the associated development methods. More specifically in the geospatial realm, Open Source components are plainly used by proprietary vendors to support their own products – but only if it does not conflict with their core business interests of selling software usage licenses. Two recent examples are the company Oracle which uses the GNU Linux operating system to run their software but not PostGIS to power their spatial database. ESRI on the other hand supports PostgreSQL (to avoid costly Oracle licenses for their customers) but not PostGIS because this would conflict with their own software product SDE. The intricacy of commercial acquisitions and their long term effects are hard to predict as can currently be seen with MySQL AB being bought by Sun Microsystems which is now coming under the control of Oracle. This shows that it is financially and strategically prudent to not rely on one vendor or product but to use Open Source and to diversify.

The Proprietary Conflict

FLOSS and proprietary software go together well, especially if they adhere to standards. That said we have to acknowledge that the business model associated with proprietary software does not go together well with Open Source. Sometimes the discussion on the pros and cons are fought out as if it were a religious war. On closer inspection the problems at hand are quite transparent and result from the deprecation of the proprietary business model which is desperately trying to compete with evolution. As we have seen the core reasons for the uptake of FLOSS are neither religious nor altruistic but simply inherent to good software development. The reason for the intermittent success of proprietary models was the absence of a ubiquitous network of communication that worked at marginal cost – the Internet. Now that we have it and know how to use it the exclusive nature of proprietary software business models has a problem.

Proprietary software needs to make all money **before** users can run the software in productive environments. With proprietary software, customers need to pay in advance and decide periodically whether to extend the maintenance contract for the upcoming contract period. With Open Source this is different. It can be run any time at no additional cost and with no commitment from a long time contract. If it does not work it can be exchanged – obviously with some cost but a lot less than what proprietary marketing wants to make us believe for so many years. Which brings us to the most obvious problem in the proprietary/FLOSS struggle: Marketing. Proprietary has too much of it and FLOSS too little. Over time a lot of Fear Uncertainty and Doubt (FUD) has been spread to the detriment of Open Source software. This has understandably caused a backlash of wild arguments against proprietary software from an as wildly marketing-unaware group of geeks. But these have organized themselves over the past years and done good work in removing most FUD so that Open Source is now socially, technologically and financially acceptable.

FLOSS will make life a lot harder for monopolists who cannot innovate as easily as an open community of thoroughly networked developers on the loose. Especially monopolists are well advised to carefully adjust their business models to this new challenge. On the good side of business FLOSS is an enabler for innovation and a door opener for start-ups and small and medium enterprises. These will also make sure that business will be more local making it more efficient and more attractive for public administrations and governments as it strengthens the local economy.

Conclusions

It can safely be said that Free and Open Source Software is here to stay. Change in large organizations has a high latency, therefore proprietary business models will be around for many years to come. Companies who employ hundreds of sales people cannot change their business model in a day. The same applies to organizations like cadastral base map agencies who operate very large and complex sets of data with high Vendor-Lock-In potential. On the other hand spatial IT also has a long tradition of using and adhering to standards because spatial data is by definition boundless and needs to interoperate. The convergence of standards and Open Source will be the core element for all future solutions.

References

[Perens, Bruce 2005] http://perens.com/Articles/Economic.html
[DoD, 2009] http://wiki.osgeo.org/wiki/Case_Studies#US_Department_of_Defense

3. Open Source Geospatial Software and the birth of OSGeo

Author: Arnulf Christl (President OSGeo)

Introduction

Free and Open Source Software experienced a strong uptake in the early days of the Internet, most notably through the GNU Linux operating system and Apache web server. But beyond these well known projects, Open Source is still a new concept for many people. Interestingly, Open Source has a long history of leadership especially in spatial software development. In the late 1970s development of the Map Overlay and Statistical System software (MOSS [1]) started after researching into existing code that was available as public domain. In the 1980s development of the Geographic Resource Analysis Support System (GRASS [2]) started, a project implemented mainly in C that over many years grew to over half a million lines of code.

With the rise of proprietary software the tendency for Vendor-Lock-In increased dramatically because vendors implemented closed formats. Therefore in the early 1990s the need for openness shifted away from developing software to making data formats interoperate more easily. To support this effort the GRASS community changed its focus and founded the Open Geospatial Foundation (OGF). Development of the GRASS software diminished and eventually subsided. But the emerging structures of the Internet allowed the project code to stay available, even although in a dormant state. In 1994 the OGF was transformed into the Open GIS Consortium (OGC [3]) later renamed the Open Geospatial Consortium to address the needs for standards by a growing global industry. Nowadays the OGC is the principal consortium for open standards in the geospatial world and works on ISO standards with geospatial relevance through a class A liaison with the Technical Committee 211 (ISO TC 211).

With the emergence of the Web as ubiquitous communication network on the Internet in the second half of the 1990s GRASS was reawakened by academia under the lead of Markus Neteler. GRASS development picked up speed again and started to grow a highly committed community which recently celebrated the 25 year anniversary of GRASS.

At the same time the first versions of the MapServer [4] software emerged in the ForNet project. It was funded by NASA in 1996 and was initially developed by Steve Lime, a single developer who included the work on shapelib [5], implemented by another early contributor, Frank Warmerdam. It soon became apparent that this new type of web

[1] http://www.scribd.com/doc/4606038/2004-Article-by-Carl-Reed-MOSS-A-Historical-perspective
[2] http://grass.osgeo.org/
[3] http://opengeospatial.org/
[4] http://mapserver.org/
[5] http://shapelib.maptools.org/

based software addressed the needs of a growing community of GIS users who recognized the potential of the Web (much later they would be known as Neo-Geographers).

In another parallel effort OGC members created the Web Map Server standard (OGC WMS) towards the end of the 1990s. Nowadays this is the standard open interface to an immense diversity of map services world wide. With the emergence of the Web 2.0 and a growing sense of belonging of the hitherto disconnected developer communities of GRASS, MapServer and several other projects the need for a common organization was articulated. The OGC was not suitable to develop or maintain software because its structures had solidified around open standards and additionally the needs of mostly proprietary vendors who then would have become direct competitors. During this time, active users and developers of the geospatial Open Source community started discussions which in the end lead to the founding of the Open Source Geospatial Foundation as is described below in more detail.

The Challenges of Open Source

Diversity and high turnover are essential to the success of Open Source. At the same time they are the root for two major challenges:

- Anyone can publish anything under an Open Source license. There is no inherent quality control in Open Source.
- There is no single, compelling reason for continuity in an Open Source project.

Both challenges also have to be addressed by proprietary software because they are not integral components of the proprietary business model. But the proprietary business model relies entirely on trust and reputation. Therefore brand quality is one of the most important assets for a proprietary software vendor. The second is continuity which is the only leverage to maximize the return on investment. So even although the motivation of proprietary businesses to address these challenges is not based on the customer's needs at least they get addressed. Open Source in itself can be based on an intrinsic motivation that will take care of both continuity and quality but it is very hard to evaluate from the outside of a project.

The web based development platform SourceForge.net[6] hosts 230.000 Open Source software projects (February 2009). The requirements to be accepted into SourceForge are that the code has to be published under a commonly accepted Open Source license and that some code is published through the SourceForge code repositories. There is no other quality assurance. The SourceForge code repositoriy is just one web based development platform, there are an additional unknown number of projects that are simply "released" under the Open Source label, sometimes with legally dubious licenses or without specifying any licenses explicitly at all.

This demonstrates the importance of looking at code quality, project governance and license model separately prior to relying on any Open Source software. Just because a

[6] http://sourceforge.net/

software is published under an Open Source license does not automatically mean that it is good software. Anybody can publish anything and postulate that it is the best around. In many Open Source projects there is no responsible legal entity to the project beyond the individual contributor.

Especially large organizations sometimes require that products which become structural core components of their IT infrastructure are backed by a reliable legal entity. Many small Open Source projects lack this legal background which excludes them from being used in these organizations.

Organizations Supporting Open Source

There are three distinct types of organizations that support Open Source for different reasons. The first two are commercial for-profit businesses and the public administration. The third type are community driven non-profit organizations whose mission is to further and promote Open Source in distinct domains.

Commercial businesses and public administrations scale – meaning that there are many instances that use and also support all kinds of Open Source. They have both specific interests but also inherent limitations to Open Source projects as we will see below. These limitations can be compensated for by non-profit organizations.

Free and Open Source Software is widely used by companies delivering commercial services as can be seen in the service provider director of OSGeo where ~~they list~~ product support, maintenance and training services are listed[7]. The Free Software license model does not conflict with their business interest; instead it is a comfortable means to enhance revenue by delivering solutions that do not incur external fees. In some projects it can even facilitate cooperation between competitors who collaborate on non-differentiating software. It is in the interest of these businesses to keep the Open Source projects alive by supporting them through sponsorship or in-kind contribution and collaboration on development.

But the support will mostly be limited to the area where it is directly profitable to the business and also vary according to the overall financial situation of the company. In a difficult financial situation this type of sponsorship is top on the cut back. Widespread commercial use of Open Source and the corresponding support will only start when the software has reached a mature state. There is little or no incentive in supporting Open Source projects in their infancy.

Public administrations can develop a specific interest in supporting Open Source because it can prevent Vendor-Lock-In situations. Larger governmental institutions have in the past often maintained development teams and implemented software on their own. Since the 1990s internal development budgets have been cut back considerably to reduce costs in the hope that "Commercial off the Shelf" (CotS) software would fill the gap. In parts this has worked out but in the long run created a much higher dependency, also because business mergers and acquisitions have led to monopolistic structures.

[7] http://www.osgeo.org/service_provider_directory

Currently a growing understanding of the inherent advantages of Open Source especially for the public administration has fueled a renaissance of Open Source support. But the years of neglecting internal capacity building can be seen, they must be compensated in order to be able to profit from all the advantages of Open Source. Additionally the working conditions should be enhanced to be able to recruit a (young and) creative workforce. Budget cuts should not simply be accepted, there are many good reasons to create, maintain and provide access to spatial data in public administrations. These should be laid out clearly to policy decision makers. Especially geospatial experts have the expertise to actually collect and convey these reasons.

Just like commercial entities the public administration has little interest in support fledgling projects that have a high risk of not succeeding. Public procurement processes have been modified in the late 80s and 90s of the last century to better address the needs of proprietary businesses, now they are not well prepared to support Open Source. Some change is already taking place, even at highest levels as can be seen in the Department of Defense of the United States of America[8] or the European Union with the Open Source Observatory Repository (OSOR[9]). But it is a long and slow process that also has to deal with the fears and resulting resistance of deprecating but still strong proprietary business interests.

For all of these reasons community-driven non-profit organizations have emerged in all major IT sectors to cater for the needs of Open Source projects that cannot be addressed by commercial businesses or public administration.

Community-Driven Non-Profit Organizations

The best organizational structure to address the challenges of Open Source are community driven non-profit organizations. They can ensure that Open Source projects prosper and develop in the most effective way. They can protect them from potentially harmful proprietary interests and give them a long term technical and legal framework. They can be home for a creative and vibrant community which will maintain and further a diverse ecology of quality Open Source software. They are also a framework for the orientation of potential users who can later turn into a new type of investor. These are not capital investors with a single interest of maximizing return on investment but with an interest of adding functionality, quality and longevity to software that helps to solve their day to day problems.

One of the earliest professional Open Source software packages that were broadly deployed was the Apache http server. It caught the interest of IBM who had come to the conclusion that the Open Source software was superior to what they were developing. This spawned the need for a legal organization for the developer community that had formed around this software and eventually gave birth to the Apache Foundation. The Open Source Geospatial Foundation was born in a similar way.

[8] http://wiki.osgeo.org/wiki/Case_Studies#US_Department_of_Defense

[9] http://www.osor.eu

In 2005 the hitherto thoroughly proprietary company Autodesk made an unexpected move by releasing their re-engineered Autodesk mapping software MapGuide as Open Source under the GNU LGPL license. The launch was supported by the developers around the MapServer project who shared a common interest in building a non-profit organization for the geospatial domain. It soon became apparent that other Open Source Geospatial software communities (for example from the GRASS project) recognized the need for an overarching non-profit organization to help stabilize development, organize conferences, create a legal background and control the governance of projects – especially because the demand for Open Source software was growing. The result was the formation of OSGeo.

The Open Source Geospatial Foundation

In February 2006 the Open Source Geospatial Foundation[10] – in short OSGeo – was founded by leading individuals from geospatial Open Source software projects. OSGeo has developed into the leading voice for Open Source in the geospatial domain. It is broadly inclusive because it is not driven by a single business with exclusive commercial interests but by a broad and diverse community of users, businesses, scientists and universities. OSGeo is home for projects implemented in different programing languages, for different user audiences and a variety of interests as such it is a community of communities.

The Mission and Tasks of OSGeo

The mission of OSGeo is to create and maintain a diverse ecology of highest quality Open Source software for the geospatial domain. The goals are to provide a stable environment for collaborative software development, a freely accessible curriculum and to promote free access to spatial data.

To achieve these goals OSGeo provides resources for existing Open Source communities and new software projects including the technical infrastructure, legal advice and financial backing. OSGeo supports FOSS4G (the acronym for Free and Open Source Software for Geospatial) on a global scale and to this end also organizes outreach and promotion. The main task in this area is to build local capacities for conferences, promote OSGeo at trade fairs and to facilitate inter-project communication. OSGeo members work on a comprehensive curriculum to help educate domain experts instead of "brand-specialists".

All these activities combine to build a solid market for business which can then in turn solidify the financial basis needed to perform these tasks. This creates a chicken and egg situation which in the first years could only be overcome with the help of many volunteers spending uncountable hours of work in their spare time.

[10] http://www.osgeo.org/

The Formal Structure of OSGeo

The structure of OSGeo has been gleaned from the Apache model and adopted to cater for the special character of binding existing communities that have already evolved in the geospatial domain. OSGeo is based on volunteer work and funded through sponsorship.

As a legal entity OSGeo needs some formal structure (see: Illustration 1: OSGeo's Structure). In a nutshell, OSGeo is owned by 94 elected charter members who are extended by 20 members each year. The charter members nominate and elect 9 directors who appoint the president. The board also appoints the Executive Director who takes care of formalities like handling finances, signing contracts in the name of OSGeo and communicating with sponsors.

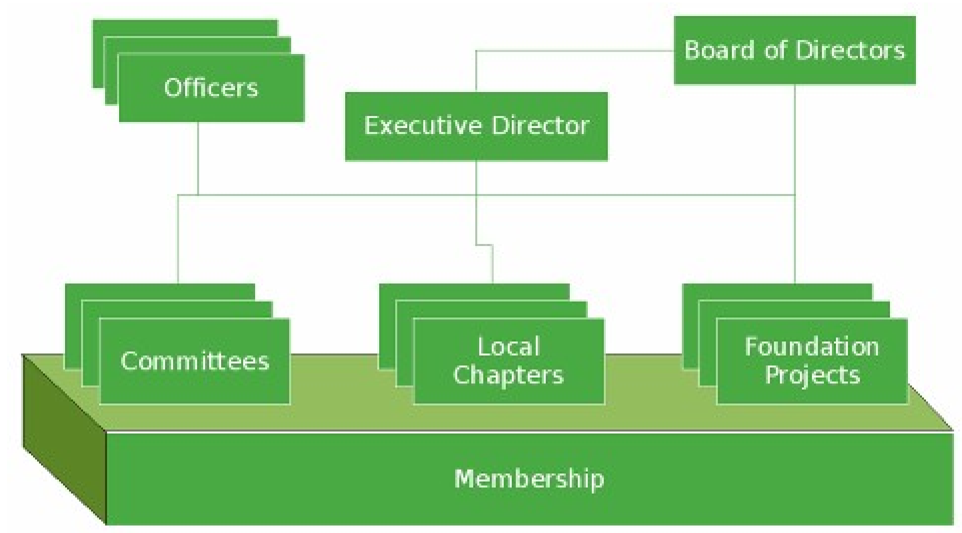


Figure 2.1: OSGeo's Structure

The board approves the budget and helps to acquire funds, for example by inviting sponsors (who act as investors) and by promoting the FOSS4G conference, which has in the past been one of the major sources of income for the organization. Formal committees are created to address different topics, each of them has a list of members who vote for a chair who then becomes an officer and Vice President of OSGeo. There are committees for outreach, conference, web site, system administration of the OSGeo services, education, finance and free geospatial data as well as the Incubation Committee.

The OSGeo Incubation Committee

The first operational committee within OSGeo was the Incubator. It addresses the need for quality assurance that is missing in many Open Source projects as was mentioned above. The Incubation Committee is formed by a broad group of developers from different backgrounds; programming languages include C/C++ and Java as well as web based technologies like PHP, JavaScript, Python and others.

Projects who are interested to join OSGeo first have to apply for the official incubation process[11]. One core criteria for acceptance is that all project code must be released under a license that has been legally confirmed by the Open Source Initiative[12]. After a mentor who guides through the process is assigned the project can be accepted into incubation. Then the work starts: the source code is checked for license and copyright consistency and the governance of the project is evaluated to prevent monopolistic structures. Basic requirements for professional development have to be met including the use of code repositories, bug tracking systems and so on, all to assure highest quality.

To graduate from incubation the project needs to abide by the Open Source rules of OSGeo as set forth in the Incubation documents. After the assigned mentor signals that the project is ready to graduate the incubation committee will scrutinize the project as a whole and eventually vote and approve for graduation. The last step to become an official OSGeo project is the approval of the board of directors.

Once accepted as an official OSGeo project users can be sure that the legal, organizational and technical structures of the project are healthy. Additionally, OSGeo will make sure that projects are provided with all that is needed to keep them going. Even in the case that a project falls out of use, the OSGeo community will make sure that there is a viable exit strategy that can be tailored to fit with the user's change management. This way the continuity and reliability of software from the Open Source domain is much higher than any proprietary company can possibly achieve. This is also the reason why many proprietary companies start to use OSGeo software in their proprietary products, a process that would not have been thinkable a few years back.

The Core of OSGeo

The core of OSGeo have always been the members of the communities that grew around the software projects. It is very simple to become a member of OSGeo, initially there is no formal process. Anybody can become a member, users, developers and academia, with a commercial, professional or hobbyist background. Becoming a regular member involves nothing more than creating a user account. The account can later be used to file tickets in the OSGeo repository, get write access the SVN repository, edit the web site and other services like shell accounts on test and build servers. Due to historical reasons the Wiki still requires a separate account; it is the place where many leave some personal information and a link to their contact data on other social networks. There are three types of membership:

- Participants collaborate on mailing lists, the Wiki, use and maintain the ticket system and work on the software stacks. This level of membership involves minimal authentication based on a valid email address.

[11] http://wiki.osgeo.org/wiki/Incubation
[12] http://www.opensource.org

- Regular members will sign up for mailing lists and become actively involved by working in committees. Usually it requires some time with active involvement in the corresponding project to become a member and vote on motions. As committees and projects are largely self organized the process to be accepted can slightly vary.
- The third category comprises the charter members. They own and control the foundation by voting for the board of directors from their midst. Currently the charter membership consists of 94 individuals from all walks of life[13]. Charter membership is renewed and extended on a yearly basis and anybody can be nominated as a charter member.

The diversity of membership also reflects in OSGeo's projects, several implement similar or even the same functionality resulting in an internal competitive situation. But this is not perceived as a problem but instead as supporting healthy diversity. A new term has been coined to convey what this means by merging cooperation and competition into "coopetition". The result is highest quality, performance and stability. One example where the concept of coopetition can be seen is the annual "Map Server Shootout" during the FOSS4G conferences. The shootout is a friendly competition of different map server projects and takes place every year. As it turns out the Java GeoServer and the C++ MapServer software are the fastest OGC WMS implementations around.

Local Chapters – OSGeo's Local Communities

One of the most important community aspects of OSGeo are over 40 Local Chapters. OSGeo Local Chapters are groups who share OSGeo's goals either in a common geographic region or through a common language or culture. Their status can be very different, some are legal entities, maintain their own funds, organize conferences, appear at trade fairs and give presentations at industry or scientific events. Others have a more informal character and provide a setting for people to meet locally or have the character of working groups with domain specific interests.

All of them share the intent to bring OSGeo into the local context, promote Free and Open Source software in the geospatial domain, localize documents and software and maintain local web sites or help maintain the main website in several languages. One of the prime interests of OSGeo is to connect with existing and emerging local communities and to support them with their local interests.

One big driver for Local Chapters is the recurring annual global FOSS4G conference. It already took place in Switzerland, Canada, South Africa, and Australia and will take place September 6th to 9th 2010 in Barcelona, Spain. As OSGeo has signed a memorandum of understanding with OGC to collaborate on standards, we will see amongst other aspects recurring interoperability experiment on standardization at future FOSS4G conferences.

[13] http://www.osgeo.org/charter_members

Conclusions

OSGeo supports Free and Open Source Software in the spatial domain and is a stabilizing factor in today's highly dynamic software and business ecology. You can participate and profit from this community of spatial expertise by using the software; secure in the knowledge that it is free of potentially disruptive proprietary interests. OSGeo is the common roof for projects and communities, a platform to create and share software, information and know-how. This also ensures highest quality software and longevity of investment. Getting involved in OSGeo and OSGeo software projects means to benefit from a highly motivated expert community.

The future of OSGeo sees continued steady growth that aims as longevity and stability instead of fastest possible expansion. With growing revenues from sponsorship OSGeo hopes to be able to contribute more to the production and maintenance of a geospatial core curriculum that does not depend on one type or even brand of software but aims at educating geospatial experts. This will help businesses to continue building up a qualified workforce and organizations to focus on solving real world problems with geospatial software instead of solving the problems of the software.