



Annual Report for 2011

OSGeo Journal Volume 11 - November 2012



Table of Contents

Editorial

04

Volunteer Recognition

07

News

08

2012 Annual Report: Software Project Annual Report Items

deegree Software Project Annual Report

11

GeoMoose Software Project Annual Report

13

GRASS Software Project Annual Report

14

QGIS Software Project Annual Report

16

2012 Annual Report: OSGeo Local Chapter Annual Report Items

California Chapter Annual Report

18

Francophone Chapter Annual Report

20

Greek Chapter Annual Report

21

Italian Chapter Report

23

Korean Chapter Annual Report

26

PDX Chapter Annual Report

28

United Kingdom Chapter Annual Report

29

OSGeo Project Spotlight

pycsw

31

Interview

Anita Graser Interview

39

Open Source GIS Software Case Study

Applying OpenJUMP to the Management of Sanitary Sewer Networks

41

Sharing GIS Data Models

47

The SurveyOS Project KML Toolkit

Getting Started With Simple Waypoints

53

Editor's Footnote

63

Editorial

By Landon Blake

What's In This Issue

I'm proud to say we published the 2011 OSGeo Annual Report before the arrival of 2013! (The credit for this lies with the volunteers that make up the OSGeo Journal Team.) This means we should be on our regular publication schedule for next year, with the annual report for 2012 out in the spring of 2013, and a peer review edition out in the fall of 2013. I still hope to publish a peer review edition of the Journal before the end of this year as well.

Inside this volume of the Journal you'll find 2011 Annual Report items from four (4) of our software projects, up from two (2) software projects last year. We have seven (7) chapters that submitted annual report items for this volume, although we lost a report from the Poland Chapter and gained reports from the Italian and Korean Chapters. Once again, the annual report items help you get a clear view of how much OSGeo related activity is taking place around the globe.



Regular Features

We've continued a regular feature from Volume 9 of the Journal in this volume. That regular feature is an interview with an OSGeo member. Last time we interviewed outgoing executive director Tyler Mitchell. This time we interview Anita Graser, a QGIS super user and contributor.

I'm excited to start two (2) new regular features in this volume of the Journal. The first is a column that will feature case studies of the application of open source geospatial software at real world organizations. In the first installment of this column we look at how the open source desktop GIS is being applied at my own company to manage sanitary sewer networks.

The second regular feature is a column that will follow the development of a brand new open source software project. This is the KML Toolkit. The KML toolkit is written in the Ruby programming language. The concept and code for the toolkit was created specifically for the Journal and this series of articles. I wanted to introduce two (2) similar columns in this volume. One for a remote sensing library written in Python and the other an OpenLayers application for tailored OSM data editing. Those columns will have to wait for our next edition. I hope you will enjoy learning about KML and Ruby programming techniques with me as the KML Toolkit is developed. If the column is successful, and we start to see real users and contributors, I'd like to submit the toolkit as an OSGeo Labs project.

Wrapping Up

We have only one topical article in this edition. It examines the problem of sharing GIS data model design, and shares some GIS data model diagrams under a creative commons license.

This volume concludes with an editor's footnote entitled "Do we need a journal?". I've written this article in response to a very insightful e-mail

from Cameron Shorter.

What's Coming Up Next

Now that work on the 2011 Annual Report is complete, I'll immediately start collecting items for our 2012 Peer Review Edition. If you have an article to contribute, or would like to assist with the peer review for this edition, please contact me.

I'm working on a couple of short guides to writing annual report items for chapters and software projects. I hope to have this done by the end of the year, so our contributors will be able to use them to submit annual report items in January.

If you have an article on a technology related topic that you want to contribute to the next issue, please let me know. You can send a message to the OSGeo Journal mailing list, or contact me personally. My current contact information is always listed in the footer or the home page at www.redefinedhorizons.com.

The Journal Blog

I've started a new blog about the Journal. I'll be posting updates on each volume there, as well as posting articles

from each edition. Let me know if you would like to contribute to the blog.

Thanks

I want to extend my thanks, first and foremost, to all the people who took the time to submit annual report items for this issue. I know you guys are busy, and many of you would rather be writing code than writing report items. I appreciate your effort and hope you will do me the same favor again soon for the 2012 Annual Report.

The pycsw Team also put a great deal of effort into preparing, reviewing, and editing the OSGeo Project Spotlight that appeared in this issue. I thank them for their efforts.

I also want to give special thanks to Eli Adam. His review and copyediting was critical to this edition of the Journal, and I really appreciate all of his help.

Volunteer Recognition

The OSGeo Journal is created by a volunteer team of open source geospatial technology enthusiasts. Without their help and support, the OSGeo would have no media mouth piece. We'd like to thank the following OSGeo members for their continuing involvement with the OSGeo Journal Team:

Eli Adam
Daniel Ames
Helena Mitasova
Scott Mitchell
Tyler Mitchell
Jorge Sanz
Micha Silver
Barry Rowlingson
Rafal Wawer
Zachary Woolard

We'd like to give special thanks and recognition to the following members of the OSGeo Journal Team that contributed to Volume 11:

Eli Adam

Eli Adam has been using open source geospatial software for eight years, currently as a GIS Analyst for Lincoln

County, Oregon and previously in the private sector as an archaeologist. He is active in the local PDX OSGeo Chapter, GeoMoose Project, and enjoys copyediting for the OSGeo Journal.

Scott Mitchell

Scott is co-director of Carleton University's Geomatics and Landscape Ecology Research Laboratory in Ottawa, Canada. He is an Assistant Professor in Carleton's Department of Geography and Environmental Studies. His research is directed at spatial analysis in support of environmental decision making, especially in agricultural and protected landscapes. Open source geospatial software and open standards provide valuable tools to enable his group's work, as well as transparent and accessible means to develop and share new algorithms and datasets. The lab's web site is <http://www.glel.carleton.ca>, and Scott can be reached at scott.mitchell@glel.carleton.ca.

Brief News and Event Announcements from the OSGeo Community

Compiled and Written by Scott Mitchell (News Editor)

To keep abreast of OSGeo news, watch the [OSGeo news page](#), or subscribe to its RSS feed. This report includes highlights from recent months, plus items specifically sent to the News Editor.

OSGeo Governance Elections

Over the last several months elections have been held for both OSGEO Charter Membership and the Board of Directors. The [current slate of Charter Members](#) now numbers 144, located around the world. Five board members were elected for two year terms: Anne Ghisla (Italy), Jeff McKenna (Canada), Daniel Morissette (Canada), Cameron Shorter (Australia), and Frank Warmerdam (Canada). Jo Cook had to step down from her position immediately prior to

the elections, and Jáchym Čepický (Czech Republic) will serve for an interim one year term to fill the vacancy. The new board has chosen Frank Warmerdam to serve as President, Daniel Morissette to serve as Treasurer, and Michael Gerlek to serve as Secretary. Congratulations and thanks to all the new and returning Charter Members and Board Members.

Conferences and Meetings

FOSS4G Beijing 2012

Unfortunately, our international meeting in Beijing was cancelled, despite great efforts to get it in place. FOSS4G events have always depended highly on local volunteers to do most of the organization, and while there have been fantastic successes, the model has inherent risk. The OSGEO Board of Directors is investigating models to mitigate these risks for future meetings. Work is already well underway for [FOSS4G 2013 in Nottingham](#), and multiple regional events have also become increasingly important and popular parallel initiatives. As an indicator of FOSS4G/OSGeo content at "other" GIS events, one can track the

distribution of the OSGEO Live DVD, which is documented at [the OSGeo Live History Wiki Page](#).

FOSS4G Regional Events - A Sampler

We've seen a plethora of regional events using FOSS4G moniker in the past few months. This is an example of the growth in regional events mentioned above. The first ever [North American FOSS4G regional meeting \(FOSS4G-NA\)](#) took place from the 10th to the 12th of April, 2012, in Washington, DC, and attracted over 350 attendees, and 50 speakers. That event was followed up by [FOSS4G-CEE & Geoinformatics](#), in Prague, May 21-23, and again slides from the meeting can be found at the conference web site. Also in Europe, June 28 was "[OSGeo.nl Day](#)" in Velp, the Netherlands, and on the 5th of September, Nottingham UK hosted an Open Source GIS Conference. Meetings in Asia included [FOSS4G Hokkaido 2012](#) and [FOSS4G Southeast Asia](#) (July 18-19 in Malaysia), and soon we'll see [FOSS4G Korea](#) in Seoul, on October 12, and [FOSS4G India](#), October 25-26.

Collaboration

Several new agreements between

OSGeo and other industry or academic associations have been announced in the past year. Most recently, the existing agreement between the [International Cartographic Association \(ICA\)](#) and OSGeo has enabled the creation of Open Source Geospatial Labs in South America and Africa, at the [Federal University of Parana](#) in Brazil, and at the Centre for Geoinformation Science at the [University of Pretoria](#) in South Africa. These initiatives join Open Geospatial labs in Europe, aiming to spread the advantages of geospatial technology to as many as possible.

Project News

GeoMOOSE 2.6 Released:

On June 19th, GeoMOOSE 2.6 was released, with updates to the included OpenLayers version, Dogo javascript library, website and documentation, and new flexibility for customizing settings. GeoMOOSE is a web application framework that enables non-developers to create web-mapping applications using familiar tools and simple configuration.

OpenLayers 2.12 Released:

OpenLayers 2.12 made its debut on June 27th, offering a new CSS-customizable zoom control, easier configuration of map projections, tile

caching to enable offline use, CSS-based tile animation, support for UTFGrid, tile queuing, and fractional zooming for tiled layers. More details can be found at GitHub.

GeoTools 8.0 Released:

The GeoTools community announced a new major release on August 7th, with many new features. The highlights include an update to Java 6, SQL joins using the WFS protocol, temporal filters, builds using Maven 2 / 3, and a new Sphinx-generated user guide with live code examples, tutorials and build instructions. More details can be found at the GeoTools web page.

OSGeo-Live 6.0 Released:

The latest version of the OSGeo-Live Xubuntu-based bootable DVD/flash drive/Virtual machine was completed in late August, and officially released just days ago at the Open Source GIS conference in Nottingham, UK. The disc/image includes demos, installers and datasets of a wide range of open source geospatial software, including 50 preconfigured geospatial applications, with overviews and quick start guides for each of them, and a collection of free spatial data. In addition to updating all the included software, a major accomplishment in this release

was to move all the Java-based applications on the image to OpenJDK 7, since Oracle has announced that Sun Java can no longer be included in Linux distributions. There has also been a lot of work to translate OSGeo-Live documentation to more languages, and the core documentation is now included in ten languages. More information and downloads are available at the OSGeo Live web page.

deegree Software Project Annual Report

Key Accomplishments

Software development for the [deegree Project](#) during 2011 was focused on the Feature Service, Map Service, Catalogue Service and underlying core modules. Data access modules for PostGIS, Oracle, MySQL and SQL Server were created or improved, with support for complex application schemas and different access modes (such as blob and relational). Basic support for WFS 2.0 was also implemented.

In March 2011 the project released deegree 3.0.3. Version 3.1 was released in October. Version 0.4.2 of iGeoDesktop was released in July. Presentations on deegree were given at the following events:

- Runder Tisch GIS in Munich, Germany during March 2011.
- FOSSGIS in Heidelberg, Germany during April 2011.
- INSPIRE Conference in Edinburgh, Scotland during July 2011.



- AGIT in Salzburg, Austria during July 2011.
- INTERGEO: OSGeo Park in Nürnberg, Germany during September 2011.
- During the OSGeo Bolsena code sprint deegree's INSPIRE capabilities were improved, along with administration and performance enhancements. An intensive code quality review was carried out during the summer of 2011. deegree project contributors Andreas and Markus opened up their developer blogs in July 2011.
- Reijer Copier, Johannes Wilden and Andreas Schmitz have been nominated as new TMC members in October.
- Andreas Poth resigned from the PSC in November.

- A roadmap for deegree 3 releases was set up and discussed among TMC and PSC.

Areas for Improvement

The project would like to improve outreach material of all kind. It wants to further open management of the project, increasing transparency. It wants to increase the number and type of contributions to the project. All kinds of contributions are welcome from an increased number of contributors.

Opportunities to Help

The project is looking for help to test release candidates at an early stage. It also needs help to provide or improve documentation (especially by native speakers of a language) and help contributing to tutorial production process. The project is actively seeking contributions to deegree 3 development. The project also encourages users and developers to share their experiences with deegree.

Outlook for 2012

The project has the following goals for 2012:

- Hold a deegree summit as combined TMC and PSC meeting.

- Create and improve deegree security modules.

- Improve deegree project web site and other outreach materials.

- Perform an infrastructure review and renewal.

- Make the deegree 3.2 "INSPIRE" release.

- Plan a deegree day.

GeoMOOSE Software Project Annual Report

Key Accomplishments

The [GeoMOOSE Software Project](#) made its version 2.4 release in 2011. This included fixes to measure, display, and printing, and a dependency change from dbase to [SQLite](#). The project was also accepted into OSGeo Incubation in 2011 and made the migration from SourceForge to OSGeo infrastructure. As part of the incubation process the project also adopted Project Steering Committee Guidelines.

Events

A GeoMoose Project Steering Committee meeting and a GeoMOOSE code sprint were held at [FOSS4G 2011](#) in Denver.

GeoMoose projects were presented at FOSS4G in Denver and GIS-In-Action.

Areas for Improvement

A more detailed development roadmap would help show where GeoMoose is headed.

Opportunities to Help

The project could use help with the



development of an automated testing suite and updates to project documentation.

Outlook for Next Year

2012 looks to be a thrilling year for the project with lots of activity! This will include more releases.

GRASS GIS Software Project Annual Report

Key Accomplishments

The [GRASS software project](#) released [version 6.4.2RC1](#). This release included eleven (11) new modules and major changes in five (5) existing modules. (See the box for details.)

Events

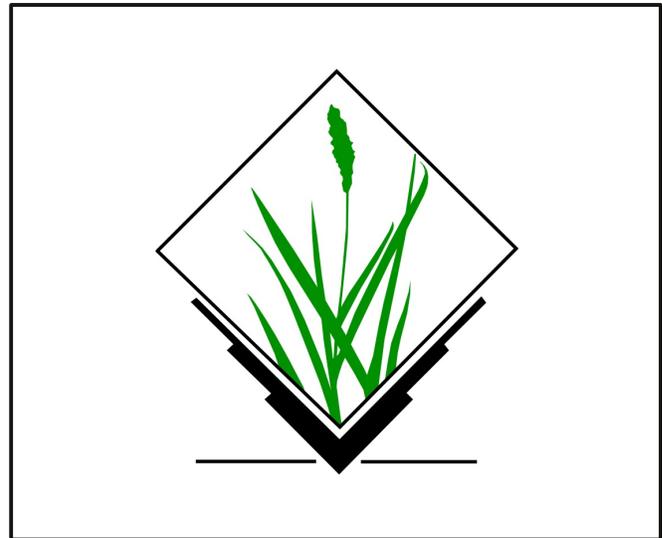
From February 9 to 11, 2011 the GRASS and [FOSS4G-IT User Meeting 2011](#) was held in Trento, Italy.

On May 12 and 13, 2011 at the [Open Software for Geodesy and Science Conference in Wroclaw, Poland](#) where lectures, practical classes and individual consultations were offered.

On May 19 and 20, 2011 talks on GRASS were given at the [International Conference on Free Software and Open Source in Geoinformatics, in Prague, Czech Republic](#).

From May 20 to 25, 2011 the [GRASS GIS Community Sprint 2011](#) was held in Prague, in the Czech Republic.

From June 6 to 9, 2011 the Scientific



GRASS GIS 6.4.2RC1 Release

New Modules

- v.net.allpairs
- v.net.bridge
- v.net.centrality
- v.net.components
- v.net.connectivity
- v.net.distance
- v.net.flow
- v.net.spanningtree
- v.net.timetable
- i.topo.corr
- m.nviz.image

Existing Module Changes

- r.univar and r3.univar
- v.generalize
- v.out.ogr
- v.kernel
- i.rectify

Workshop 2011 "Spatial Analysis with GRASS" took place in the [Department of Climatology and Atmosphere Protection, University of Wroclaw](#).

From June 17 to 19, 2011 the first GRASS and [GFOSS Hellenic Users Camp](#) was held at Paou Monastery, in Pelion, Volos, Greece.

Areas for Improvement

The GRASS software project would like to develop a migration guide for public agencies and wants to improve the sponsorship program.

Opportunities to Help

The software project is looking for help in translating GRASS GIS messages. It also needs help preparing marketing material, including updates to flyers and posters and preparation of a new web site.

QGIS

Software Project

Annual Report

Key Accomplishments

The QGIS Software Project made the release of [QGIS 1.7.0 "Wroclaw"](#) on June 19, 2011. It also supervised three (3) projects for the 2011 Google Summer of Code. These Summer of Code projects included a SAGA interface for QGIS, a Database Manager Plug-In, and QGIS Mobile. In addition, during 2011 the project provided [a new wiki and bug tracker](#).

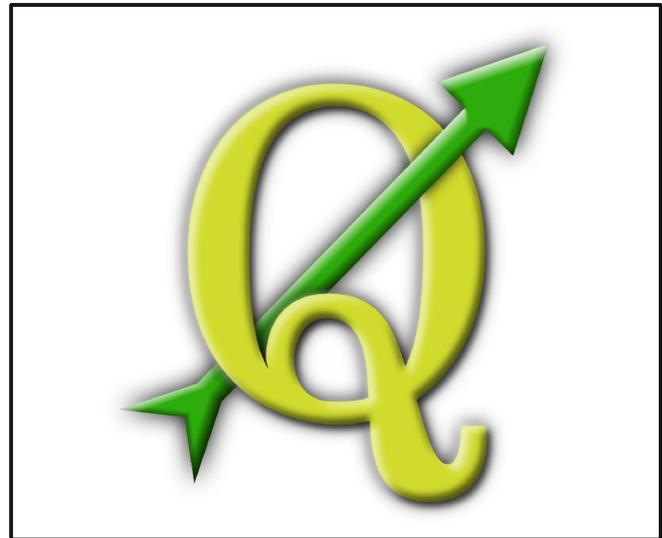
Events

The QGIS project held a number of developer meetings ('hackfests' and 'bughunting') during 2011. The purpose of these meetings is to gather people committed to improving the Quantum GIS project. These people include developers, documenters, bug reporters as well as users.

In 2011 the following developer meetings were held:

- 1) Lisbon, Portugal in April
- 2) Zurich, Switzerland in November

In 2011 the following user meetings were held:



QGIS Developer and User Meetings

Developer Meetings

- Lisbon in April, 2011
- Zurich in November, 2011

User Meetings

- Kassel in September, 2011
- Rapperswil in May, 2011

- 1) Kassel in September
- 2) Rapperswil in May

Areas for Improvement

The software project is interested in setting up a software test suite and in making improvements and consolidations to the user interface.

Opportunities to Help

The Quantum GIS project welcomes help from developers, UI designers, software testers, translators and users. Join the mailing lists or visit one of our events to learn more.

Outlook for Next Year

In 2012 the software project would like to make the release of QGIS 1.8. It hopes to again participate in the Google Summer of Code. It will continue to host developer and user meetings.

California Chapter Report

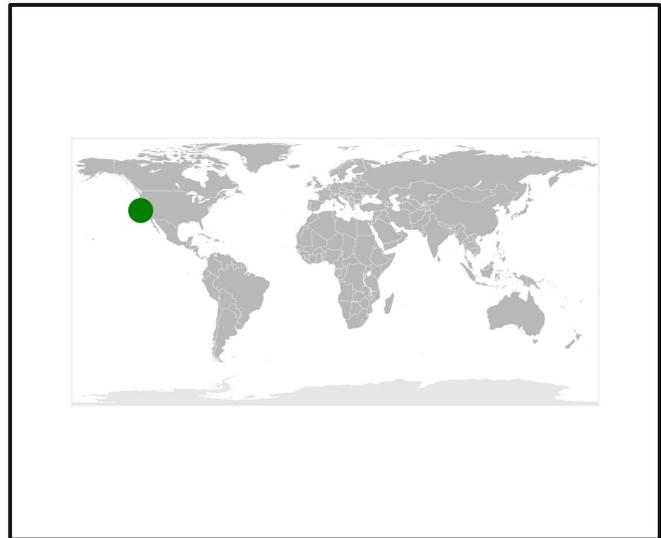
Chapter Events

On June 11, 2011 Chapter Member Brian Hamlin attended the 3 day [FLUXNET and Remote Sensing Open Workshop](#) held at the [David Brower Environmental Center](#) in Berkely, California. During the event Brian was able to distribute the OSGeo LiveDVD to many of the visiting scholars.

On July 11, 2011 Chapter Member Brian Hamlin attended the annual Meeting of the [California Invasive Plant Society](#) held at the David Brower Environmental Center in Berkely, California. During the event Brian was able to distribute the OSGeo LiveDVD and promote OSGeo to attendees.

During April 12 to April 18, 2011 chapter members managed an OSGeo booth at the [Association of American Geographers Annual Conference](#) in Seattle, Washington.

On October 22, 2011 the Chapter held its Annual Meeting in Davis, California. The morning session of the meeting included several talks. (See the box for more information.) A round table discussion on participation in open source projects was held during the lunch break. The afternoon session featured a cartography sprint, code sprint, and



2011 Annual Meeting Talks

- Quantum GIS & Inkscape:
 - Cartographic Tools for Attractive Maps (Michele Tobais)
 - A Brief Introduction To OpenJUMP (Landon Blake)
 - A Pragmatists Guide to PostGIS 2.0 (Brian Hamlin)
 - Using GRASS and R for Landscape
 - Regionalization Through PAM Cluster Analysis (Allan Hollander)
 - Analyzing Raster Data With R (Robert Hijmans)
 - Raster Formats Performance
 - Optimization/Comparison (Frank Wammerdam)
 - Python and PostGIS (Nate Roth)
 - QGIS Plugin Case Study - Geological Surveys (Bob Moskovitz)

attempted demonstration of kite photography.

On November 17, 2011 Chapter Member Landon Blake gave a talk entitled "Solving Three Problems Faced By Communities in the California Central Valley" at the [UC Berkeley GIS Day Event](#).

Member Activities

Chapter Member Ragi Burhum continued to coordinate the [San Francisco Geomeetup](#).

Chapter Member Landon Blake founded the [California Central Valley Geospatial Professionals Group](#). The group held [its first meeting on September 30, 2011](#). Guest speakers at the meeting included Annette Lockhart, GIS Committee Chairman of the California Land Surveyors Association, and Erin Mutch of Solstice GIS.

Michele Tobias gave a talk on QGIS and Inkscape at the [FOSS4G Conference](#) in September. She has also been collaborating with the [Public Laboratory for Open Technology and Science](#) on open source hardware and community science tool development.

Alex Mandel was appointed chair of the [Systems Administration Committee for OSGeo](#). He gave a guest lecture entitled "Introduction to Geospatial - the Open Source Method" in the Introduction to GIS course at UC Davis. He also taught a course on the introduction to climate

change analysis with QGIS at the [US Forest Service International Seminar on Climate Change and Natural Resources Management](#).

The course was taught with OSGeo Live, which was provided to all 25 of the participants.

Brian Hamlin provided the chapter update at FOSS4G in Denver. More than 50 Californians attended the conference.

Allan Hollander presented at FOSS4G, in the academic track.

Francophone Chapter Report

Key Accomplishments

The Francophone chapter completed the clean-up of an administrative paper needed to set up a legal association for the chapter.

It also elected a new governing board in June 2011.

Areas for Improvement

The chapter would like to encourage people to contribute more to translation projects. It wants to create a marketing package and help people in organizing OSGeo booth at events. It would like to attract additional contributors and project managers. It has a general goal of improving communication in the Francophone geospatial community.

Opportunities to Help

The chapter has four (4) opportunities to help in chapter activities. This include help in translating [MapServer](#) documentation, help in organizing Francophone QGIS events, help in monitoring a booth at several conferences, and help with chapter marketing duties.

Outlook for 2012

In 2012 the chapter would like to continue the work of setting up a legal association. It would also like to improve the document translation process. There are also plans to organize an “OSGeo-fr Day” event.

Italian Chapter Report

Introduction

The OSGeo Italian Chapter was founded in 2007, and currently has around 90 members and 9 OSGeo Charter Members.

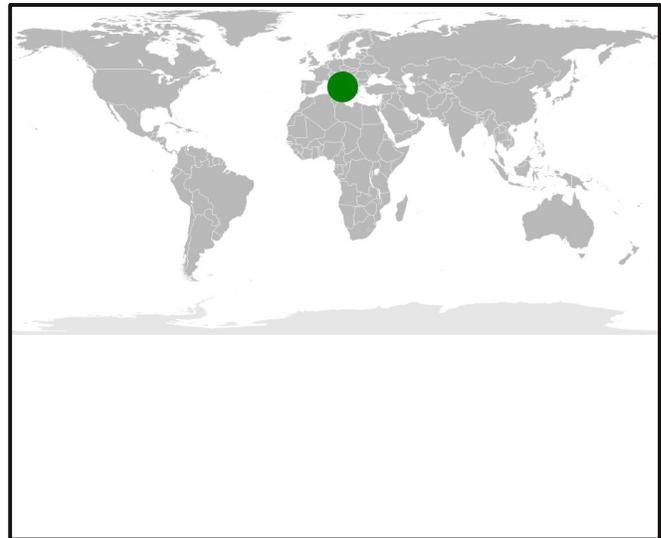
The chapter is carrying out activities related to free and open source software and about open data. In particular, the chapter supports the release of geodata from public agencies under a suitable open data license.

Key Accomplishments

The chapter carried out a number of community events in 2011. These events have been a great opportunity for discussion on new advances in open source geospatial software and on national geospatial solutions. They also provided networking opportunities for developers, researchers and users.

Events

Twelfth Meeting “[degli Utenti Italiani di GRASS](#)”: This was a three (3) day conference targeted to researchers and developers on the field of open source GIS, especially GRASS GIS. The event took place in [Trento University](#) and included, as usual, both workshops and



2011 Chapter Events

- GFOSS Day 2011
- Twelfth Meeting "degli Utenti Italiani di GRASS"
- Open Data and Documents Study Day
- OSMit2011

presentations.

GFOSS DAY 2011: This was a two (2) day conference, targeted mainly to professionals of the public and private sector. The event was held in Foggia. In the morning on the first day, workshops were held on several open source geospatial software packages. Workshops were carried out using the [OSGeo Live](#) environment, and a brief explanation about the OSGeo Live project was given to those in attendance.

Open Data and Document Study Day: On April 13th, 2011 in Bologna, the chapter organized a Study Day (Giornata di Studio) on geospatial open data. The participants included the chapter, Emilia-Romagna Region and CSI Piemonte. They produced a set of guidelines related to geospatial open data ready for application in the Italian context. The definitions in the guidelines were borrowed from [Open Knowledge Foundation's Open Definition](#). The aim of the guidelines is to encourage the creation of services based on open data from public administrations. The guidelines recommend the usage of internationally recognized licenses, open formats, and raw data access.

OSMit2011: The annual OpenStreetMap Italian conference was sponsored by the chapter and two (2) chapter members were the organizers of the conference.

Member Activities

Chapter members collected donations for several open source geospatial software packages, as well as for GRASS GIS and QuantumGIS code sprints. Its members also participated in these code sprints. Chapter members are also active in other fields and associations, such as archaeology, ecology, public administration, and local Linux User Groups. These activities allow the chapter to increase the discussion and adoption of open source geospatial software outside the relatively small circle of GIS specialists.

Opportunities to Help

The chapter manages translation efforts on various FOSS software packages, including the OSGeo-Live DVD, and is a hub for packagers for various Linux distributions. Volunteers to assist with translating and packaging work are always welcome!

The [GFOSS.it wiki](#) is open to contributions and collects useful information about Italian geospatial data, as well as tips for software installation and configuration.

Outlook for 2012

The [Thirteenth GRASS and GFOSS Meeting](#) will be held on Trieste on February 15 to 17, 2012.

At the annual meeting, chapter associates will elect the new Board of Directors and Chairman to be in charge during 2012-2013 term.

A qualified chapter representative will attend the Italian annual Conference of Free Software on Ancona, June 22 to 23, 2012.

The chapter plans to sponsor and attend the SmartCities event in Bologna on July 6, 2012.

The next GFOSS Day will be in Turin on November 14 to 17, 2012. For the first time this event will include the annual Open Street Map Italian conference (OSMit), thus signifying a strong integration between the OSM and open source geospatial communities.

Greek Language Chapter

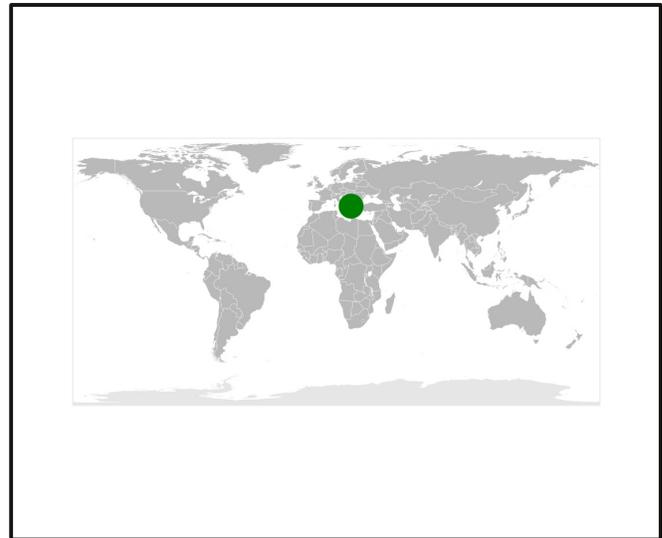
Key Accomplishments

On June 17 to 19, 2011 the chapter carried out its main event with great success. It was a joint effort organized by the OSGeo Greek Local Chapter, the Greek GI Association (HellasGI) and the Greek Open Source Association (E/ELLAK). The event was the First GRASS and GFOSS Users' Camp. It was held at the Paou Monastery, University of Thessaly, Pelion, Volos, Greece.

The event was highly successful with over 50 participants who stayed for the whole duration of the event. Many Greek and international experts (people from [GRASS](#) and [gvSIG](#) communities were invited) presented issues related to the main theme. Many participants recommended the event be repeated next year.

The chapter has coordinated work on the Greek translation of Version 4.5 and 5.0 of OSGeo Live. The Greek translation of the OSGeo Live DVD is one of the most complete.

Weekly tutorials and presentations were provided by the chapter at the



[National Technical University of Athens \(NTUA\)](#) about Free and Open Source Geospatial Software. Approximately 15 to 20 students attend each week.

The chapter has also coordinated Open Source Geospatial Software adoption in classes in universities. An example is the class at TEI of Serres with [gvSIG](#) and [GRASS](#).

Areas for Improvement

We still do not have the participation in the chapter we expected. We need to intensify the efforts to increase the membership and the overall awareness. A big challenge is securing funding for organizing further events. Efforts to translate the OSGeo web site and some of the OSGeo Software stack into Greek will definitely help bring more people to the chapter.

Opportunities to Help

In the future we will definitely need speakers for national FOSS or local GIS events. We would also like to demonstrate some cases of successful use of FOSS GIS by the public sector to local government officials.

We would like to plan some demo classes on FOSS GIS for students in universities and how these can be integrated in their curricula in order to show that we can achieve the exact same educational result as programs that use proprietary software. (This process has started to happen but more involvement is needed).

Outlook for 2012

We would also be very interested in having the ability to apply for funding as a chapter or organization and we would like to seek help and information from other local chapters that have or plan to do the same. We would also like to become an official OSGeo Chapter.

Korean Language Chapter Report

The [OSGeo Korean Language Chapter](#) was formed as an unofficial chapter in February, 2008 and changed its status to an official OSGeo Local Chapter in March, 2009. The OSGeo Korean Language Chapter had 194 members at the end of 2011.

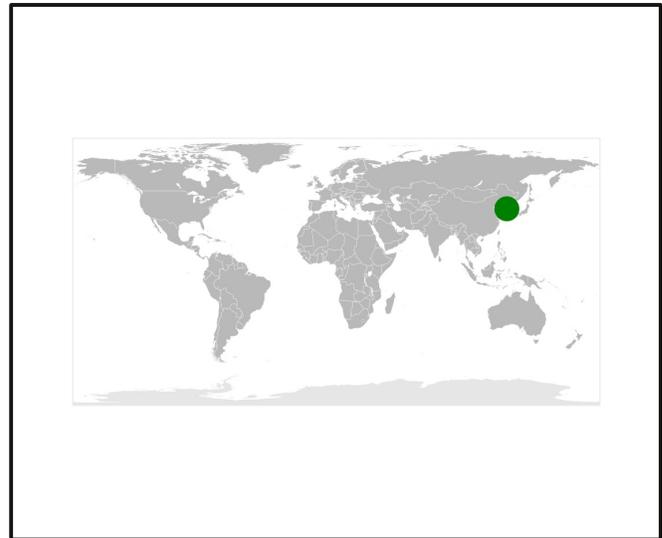
Events

On October 28, 2011 the chapter held the first annual OSGeo Korean Language Chapter Conference at KINTEX in Goyang City, Korea. Keynote speakers included Arnulf Christl, Toru Mori, Daisuke Yoshida, Byungnam Choi, and Sanghee Shin. The event was highly successful with between 60 to 80 participants, 5 keynote talks and 8 additional presentations.

The chapter held two (2) regular chapter meetings in February and August of 2011. Both meetings were held in Seoul, Korea. At the meetings chapter members shared knowledge & experiences around open source GIS. They also discussed the future activities of OSGeo Korean Language Chapter.

Key Accomplishments

The chapter carried out several localization & translation efforts. These



efforts included translation of [QGIS](#) menus & messages. Efforts to translate QGIS, [PostGIS](#), [GeoServer](#), and [OpenLayers](#) materials were also started.

The chapter has been involved in several open source projects. These included the following:

- 1) KOPSS([Korea Planning Support System](#)) Open API Project funded by KRIHS([Korean Research Institute of Human Settlement](#).)
- 2) The "Strategies on Building the Platform for Geospatial Information Technology Development : Based on Open Source Thinking" funded by KRIHS([Korean Research Institute of Human Settlement](#)).
- 3) Development of open source GIS

education program funded by NECGIS(National Education Center for GIS).

The chapter conducted several outreach activities including lecture, seminar and exchange programs. Outreach lectures were provided to LH Corp, KRIHS, KCSC([Korea Cadastral Survey Corp](#)) and other organizations. As part of the exchange program two (2) delegates were dispatched to Tokyo, Japan to give keynote speech at FOSS4G 2011 Tokyo/Osaka.

An Official OSGeo Korean Language Chapter web page was created to share our activities. It can be found at <http://www.osgeo.kr>.

Areas for Improvement

There is too much dependency on a few key members. The chapter wants to increase the number of active members. The Korean chapter is still not a legal entity, and this imposes several limitations on possible chapter activities.

The chapter would like to expand activities to an international scale. The chapter has very talented members, but their activities are somewhat isolated to Korea.

The chapter also needs to seek stable revenue sources to run our chapter. Currently the chapter finances are highly dependent on limited donation sources.

Opportunities to Help

The chapter is seeking speakers for [FOSS4G Korea 2012](#) and is looking for opportunities to host more open source GIS related events in Korea or in Asia region.

Outlook for 2012

The chapter predicts a very prosperous year in 2012. As awareness of open source GIS increase in Korea rapidly, many organizations will want to cooperate with the chapter. In 2012, the chapter hopes to attract more members and to embark on efforts to run the chapter in more systematic way.

The chapter will explore the possibility of hosting FOSS4G 2015 in Korea with another GIS related society like KSIS(Korean Spatial Information Society).

PDX Chapter Report

Chapter Events

The chapter held monthly meetings with numerous presentations. They also hosted an unconference. See the box for more information on the unconference talks.

Many chapter members presented at regional GIS conferences on OSGeo topics.

Key Accomplishments

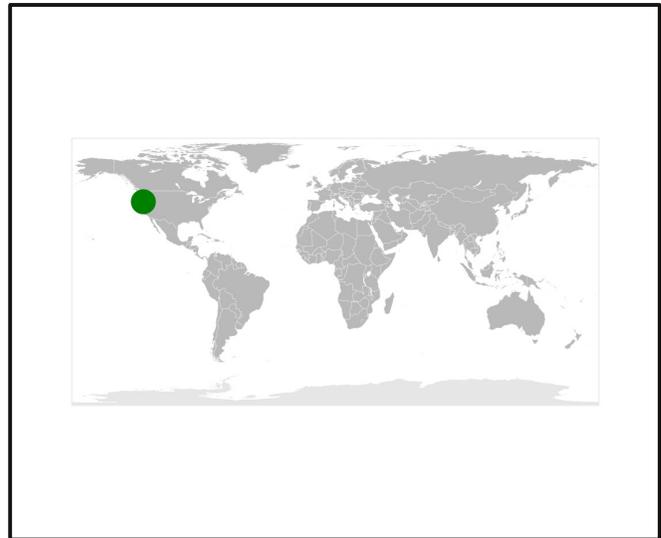
Chapter members started taking better notes and using the wiki. They finished most of the OSGeo Local Chapter checklist.

Opportunities to Help

The chapter is looking for people to join the planners list and help schedule presentations, post meeting announcements, and perform other planning tasks.

Outlook for 2012

The chapter looks forward to greater meeting attendance and the need for a larger meeting space.



2011 PDX Unconference Talks

- Civic Apps, OSM, and Authoritative Data
- Crises Mapping: rdtm.org
- State of Oregon Services and Standards
- Open Street Maps HowTo
- Building Minecraft Maps With USGS Data
- Routing In Open Source
- Can Open Source Replace ESRI
- QGIS Python Extensions
- Getting Started With R
- Integrating OpenLayers with GeoExt

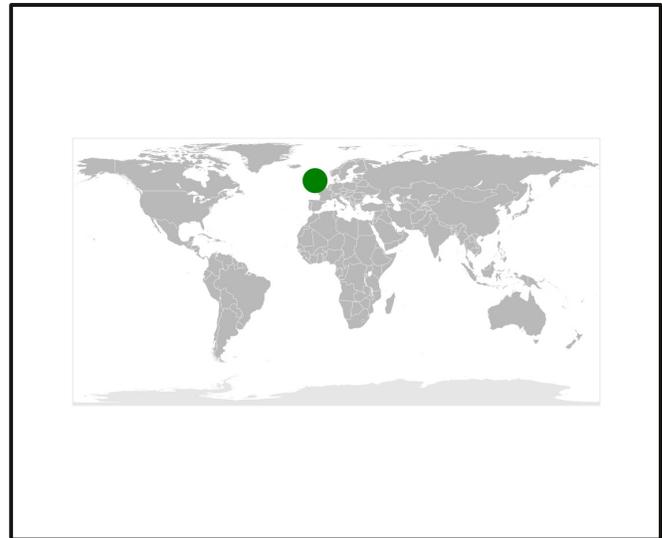
OSGeo United Kingdom Chapter Report for 2011

Key Accomplishments

2011 has been a great year for the chapter. It reached 150 members on our mailing list, up from 100 last year. The chapter is holding regular IRC meetings, which helped chapter members stay in touch. It helped to co-host the [OSGIS 2011 Conference in Nottingham](#), and chapter members presented at a number of other open source and geospatial events and workshops. This included the AGI ([Association for Geographic Information](#)) event.

The chapter has been improving the communication with chapter members and other interested parties through monthly IRC meetings and an official OSGeo:UK twitter account (@osgeouk). Members have also been busy promoting open source GIS through local Linux User Groups and other places.

The chapter has members on the board of the new [Open Source Geospatial Lab at the Centre for Geospatial Sciences in Nottingham](#), has two OSGeo Charter



Members, and the Chapter Representative was elected as an OSGeo Director.

Areas for Improvement

The chapter needs to continue raising the profile of open source geospatial software in the UK. It also needs to ensure that we take advantage of things like the UK Governments "open agenda". More chapter members and more opportunities to get together are always useful!

Opportunities to Help

The chapter is always on the lookout for new members, and new events to promote OSGeo. The chapter invites you to join the chapter mailing list and get involved!

Outlook for 2012

The outlook for 2012 is good. The

chapter's main focus is our bid to host [FOSS4G in the UK in 2013](#), which chapter members are very excited about. The chapter is also hoping to organize a joint event with Ordnance Survey, and to continue its strong association with the AGI and the [University of Nottingham](#). The chapter hopes to continue to gather more case studies about exemplar uses of open source geospatial software, and increase chapter membership to 200 people. The chapter also wishes to attract more companies to be part of the local chapter, and is investigating potential sponsorship options.

pycsw: an OGC CSW server implementation written in Python

The pycsw Development Team

Overview

[pycsw](#) is an OGC CSW server implementation written in [Python](#).

pycsw implements clause 10 (HTTP protocol binding ([Catalogue Services for the Web](#), CSW)) of the [OpenGIS Catalogue Service Implementation Specification](#), version 2.0.2. pycsw allows for metadata publishing either from its built-in data model, or through configuration. In the configuration mode the user can bind to an existing metadata model.

pycsw is Open Source, released under an [MIT license](#), and runs on all major platforms (Windows, Linux, Mac OS X).

History

pycsw is a young project. Initial development began in 2010, with the main focus being to provide a very lightweight Python CSW server solution (in comparison to many Java-based



CSW servers).

Another goal was to provide a standalone CSW server implementation. This means that metadata is created and managed and updated elsewhere, and pycsw thus acts as the publishing component for geospatial resource discovery.

Another focus was extensibility: OGC Catalogue Service, by design, allows for the definition of application profiles to support additional metadata formats (the core metadata model is Dublin Core + ows;BoundingBox), so this was an important design decision prior to initial implementation. Features (such as metadata formats, encodings, and harvesting) are implemented such that they can be extended to provide additional variations of a given feature.

The OGC CITE tests were extensively used to establish a benchmark of compliance to the specification (pycsw is not an OGC approved compliant product, but does pass all the CITE tests).

Version 1.0.0 was released in 2011, providing initial support for basic CSW operations, ISO Application Profile and INSPIRE Discovery Services support. [SQLite](#) and [PostgreSQL](#) were the initially supported databases.

In 2012, version 1.2.0 was released, adding additional search interfaces (SRU implementation, [OpenSearch](#)). [MySQL](#) database support was also added in this version, providing further flexibility for deployment. WMS harvesting and GeoNode support were also implemented, as well as the ability to provide [JSON](#) output of search results.

In September 2012, version 1.4.0 was released. This release brought many important features, including WSGI support. The WSGI development enabled pycsw to be integrated into existing Python frameworks, such as [Django](#) or [Flask](#).

The first formal deployment of pycsw was by [INSIDE Idaho](#). INSIDE Idaho is

the official geospatial data clearinghouse for the State of Idaho. INSIDE Idaho serves as a comprehensive geospatial data digital library, providing access to, and a context within which to use, geospatial data and information by, for, and about Idaho.

What makes this deployment interesting is that pycsw, developed in a Linux / Apache environment, was deployed in a Windows / IIS environment.

Design

From its inception, pycsw strived to be lightweight, flexible, easy to install and deploy. A typical install takes less than 10 minutes.

Simple Configuration

Configuration is governed by a simple configuration file using the familiar Windows INI syntax. Users simply edit this file as required and changes are reflected in server behavior. Python supports this format natively with its ConfigParser standard library. In addition, those with customized applications can generate their own ConfigParser object (e.g. from an external configuration, Python dictionary or database) and send to pycsw in the same fashion. XML

configuration files were considered early on in development, but it was decided for performance reasons to stick with a simple, plain text format.

Repository

Metadata is handled by way of a "Repository", which is defined as a physical database instance along with advertised queryables. CSW requires the advertisement of specific queryables, and advertises these via mappings to their underlying columns and properties in the database.

Dispatcher

All requests are performed via HTTP GET or HTTP POST. The dispatcher is vital in deciphering what the interface of the request is (CSW, SRU, OpenSearch, etc.), and responds with the appropriate headers and payload.

Plugin Architecture

A plugin architecture is provided for developers to extend the codebase in order to support additional application profiles, formats, and repositories.

No XSLT

To avoid the overhead of XSLT processing, pycsw was designed to handle metadata by processing XML elements into a relational model (database). Thus there is almost no

transformation of XML within the codebase. Metadata XML is generated from the database.

Features

As of current writing (October 2012), pycsw implements the following features:

- Fully implements OGC CSW 2.0.2
- Fully passes the OGC CITE CSW test suite (103/103)
- Implements INSPIRE Discovery Services 3.0
- Implements ISO Metadata Application Profile 1.0.0
- Implements FGDC CSDGM Application Profile for CSW 2.0
- Implements the Search/Retrieval via URL (SRU) search protocol
- Implements OpenSearch
- Supports ISO, Dublin Core, DIF, FGDC and Atom metadata models
- CGI or WSGI deployment
- Simple configuration
- Transactional capabilities (CSW-T)
- Flexible repository configuration
- GeoNode connectivity
- Open Data Catalog connectivity
- Federated catalogue distributed searching
- Realtime XML Schema validation
- Extensible profile plugin architecture

Technology

pycsw is written in Python and leverages the following technologies:

[lxml](#) is used for XML request parsing and validation, as well as serializing XML responses. lxml is a cornerstone technology used in the codebase. lxml is the Python binding to the libxml2 C library.

[Shapely](#) is used for handling spatial predicates in an independent manner. It was decided to use Shapely to be able to deal with geometries agnostic to a given database environment. This allows pycsw to bind to any database. OGC Well Known Text (WKT) is used as the internal geometry format. EPSG:4326 is used as the internal coordinate reference system. Shapely is the Python binding to the [GEOS C library](#).

[SQLAlchemy](#) is used as the database abstraction layer and provides a Pythonic approach to working with databases (as opposed to raw SQL scripting).

[OWSLib](#) handles the heavy lifting of parsing XML formats and interacting with OGC Web Services for harvesting.

pyproj is used to handle coordinate transformations, e.g. for CSW requests

which provide non-geographic coordinates. pyproj provides Python bindings to the proj.4 C library.

INSPIRE Support

Early versions of pycsw supported only core CSW 2.0.2 and were able to complete OGC CITE tests with a 100% success rate. There was an initial goal of the project to be able to work with plugins, in a way that would make it very extendable and easy to configure. For example, the end user should be able to install or uninstall a plugin by adding or removing a folder within the plugins directory.

The first metadata profile selected to be implemented was the APISO profile of CSW 2.0.2 since it was widely used in current CSW implementations. This profile was ready and stable for version 1.0.0 of pycsw.

Around that time, there was a growing interest in Europe about Catalogue Service implementations that would support the draft guidelines of the INSPIRE Discovery Service specification. This specification was not final at that point. However, through the INSPIRE Directive there was a final decision about the specification of metadata. This specification supported ISO 19115 and 19136 with some additions and

profile modifications.

Just before pycsw 1.0.0 was released, Version 3.0 of the INSPIRE Discovery Service specification was the first to leave beta and become official. At this point pycsw supported multilingual metadata, as well as additional service metadata needed to comply with the directive. This was implemented within the APISO profile and not in a separate profile (INSPIRE is heavily based on APISO and would make a new profile highly redundant). The APISO plugin was configured to have INSPIRE support by setting a configuration switch. The user would then provide the additional metadata needed by the INSPIRE specification. The final specification for Discovery Service brought some changes and a new XML namespace to follow. This was added and fully supported in pycsw 1.0.0.

The pycsw database model supported the mandatory fields of INSPIRE for APISO and search/read/write/update procedures. This support was enabled by enhancements made to the underlying OWSLib ISO 19139 implementation. Those patches were contributed back to OWSLib.

Integration with Python Frameworks

pycsw has the ability to operate in 'library' mode by external applications. It is also possible to use pycsw within your application through pure Python request and response (no HTTP) mechanisms. This allows for easy integration and inclusion of a CSW into an existing website or application and information model.

GeoNode

GeoNode is a platform for the management and publication of geospatial data. It brings together mature and stable open-source software projects under a consistent and easy-to-use interface allowing users, with little training, to quickly and easily share data and create interactive maps. GeoNode provides a cost-effective and scalable tool for developing information management systems.

As part of the GeoNode 2.0 roadmap, development was undertaken to abstract CSW functionality to any CSW (pycsw, GeoNetwork OpenSource, deegree), which allows flexibility in cataloguing in GeoNode. In addition, GeoNode was updated to deploy pycsw inline as a WSGI application and working directly off the GeoNode database (Django models), thus reducing redundancy in metadata

management and software deployment footprint. pycsw is the default CSW server for GeoNode 2.0.

Open Data Catalog

Open Data Catalog is an open data catalog based on Django, Python and PostgreSQL. It was originally developed for <http://OpenDataPhilly.org>, a portal that provides access to open data sets, applications, and APIs related to the Philadelphia region. The Open Data Catalog is a generalized version of the original source code with a simple skin. It is intended to display information and links to publicly available data in an easily searchable format. The code also includes options for data owners to submit data for consideration and for registered public users to nominate a type of data they would like to see openly available to the public.

In the same spirit as GeoNode, pycsw was chosen as the CSW server for Open Data Catalogue. Given the ODC's desire to have an interoperable CSW as part of the project, and pycsw's ability to integrate with Django, pycsw was a natural fit for the CSW requirements of the project. As in the GeoNode scenario, pycsw uses ODC's database directly via Django models.

OSGeo Involvement

OSGeo Live DVD

pycsw is available on the OSGeo-Live project. The overview and quickstart provides further information on using pycsw in OSGeo-Live.

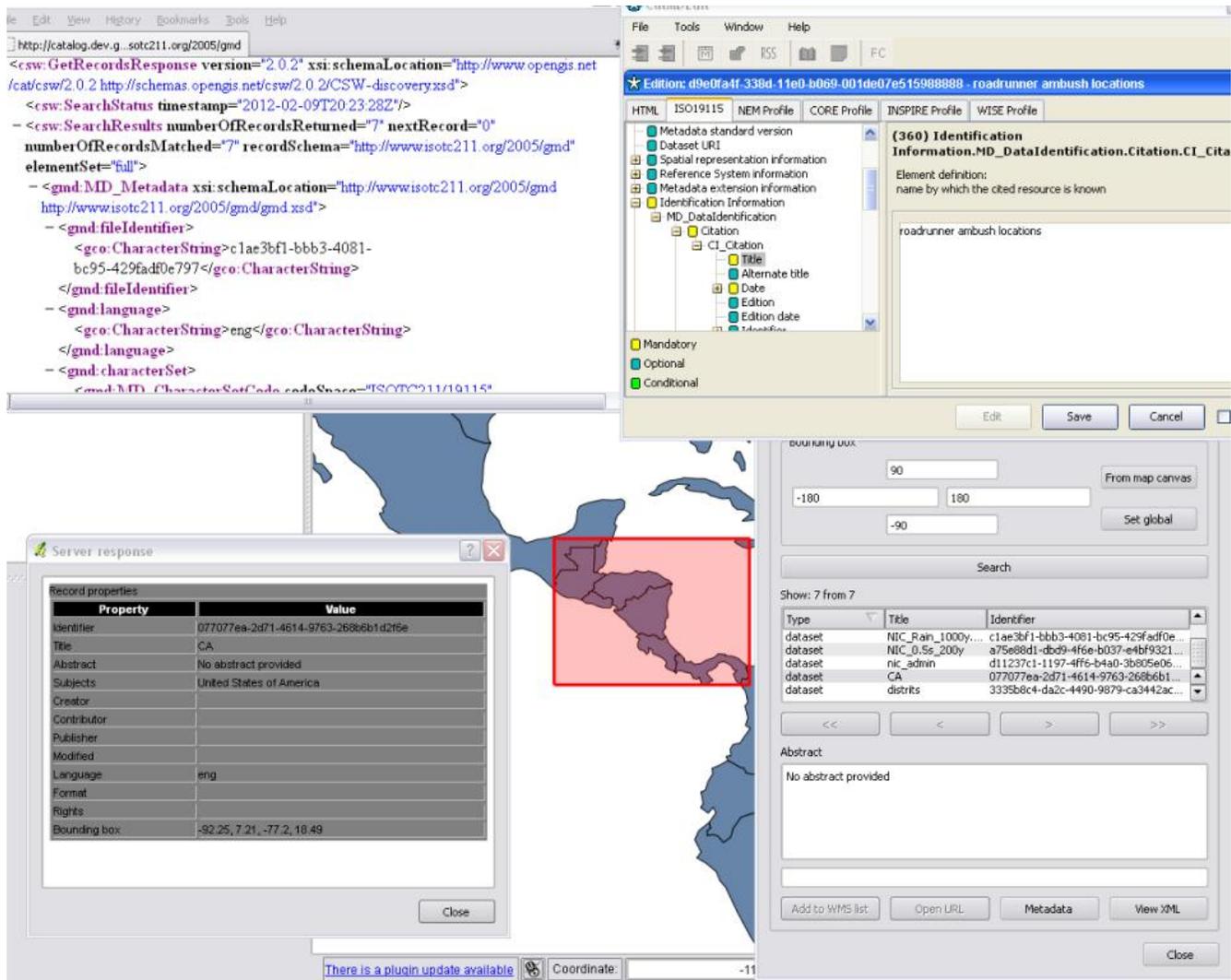
Labs/Incubation

The project is currently in OSGeo Labs and leverages various OSGeo infrastructure, with the goal of reaching incubation status and becoming an approved OSGeo project.

Future Development

A few areas of interest for future development include:

- Spatial database and geometry object support (PostGIS, MySQL spatial): currently pycsw works from a WKT string column and Shapely for spatial operations. Adding support for true geometric objects would enable using direct spatial support. In addition, for non-spatial databases, WKB is being considered as a means to improve performance.
- Search engine support: Currently pycsw works in a similar manner as an OGC Web Feature Service (WFS).



pycsw Screenshot

Adding support for true search engine libraries would enable full text indexing and search result relevance.

- Web administration/metadata editing: Providing a front end administration tool would allow for more user-friendly interaction to load metadata, edit server settings, etc.

- OGC Catalogue 3.0: Future versions of the OGC Catalogue Service Specification will require pycsw to support multiple versions (the codebase is currently bound to 2.0.2).

Some of the abovementioned features will come at the cost of ease of installment and deployment, and will be developed as optional, configurable

components.

Conclusion

pycsw is a young, lightweight, flexible and fast CSW server implementation. Already the project is being used as a standalone service and in various applications and will hopefully expand in usage over time.

Community involvement is welcome from users and developers. Please feel free to visit <http://pycsw.org>.

References

D. Nebert, A. Whiteside, P. Vretanos (2007)
Open Geospatial Consortium -
Catalogue Service for Web.
OGC 07-006r1 pp.204.

U. Voges, K. Senkler (2007)
Open Geospatial Consortium Inc. ISO
Metadata Application Profile
OGC 07-045 pp.125.

Directive 2007/2/EC of the European
Parliament and of the Council of 14
March 2007
<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32007L0002:EN:NOT>

Commission Regulation (EC) No
1205/2008
<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32008R1205:EN:NOT>

Technical Guidance for INSPIRE
Discovery Services
http://inspire.jrc.ec.europa.eu/document_s/Network_Services/TechnicalGuidance_DiscoveryServices_v3.1.pdf

Tom Kralidis
<http://kralidis.ca/>
tomkralidis AT hotmail.com

Angelos Tzotsos
National Technical University of Athens
<http://users.ntua.gr/tzotsos/tzotsos AT gmail.com>

Adam Hinz
Azavea
<https://github.com/ahinz/hinz.adam AT gmail.com>

Interview With Anita Graser

In this volume of the journal we interview Anita Graser. Anita is a long time advocate of open source geospatial software. She is also a user and developer for [QGIS](#). In the interview we ask Anita how she got involved with GIS, learn a bit about GIS, and conclude by getting her thoughts on the OSGeo as an organization.

OSGeo Journal:

Where do you currently live and work?

Anita:

I live in Vienna, Austria and work at the [Mobility Department of the Austrian Institute of Technology](#).

OSGeo Journal:

When and how did you first get involved in GIS?

Anita:

I always loved maps and tinkering with computer scripts. When I found out that you can actually combine those two things, I went to the university to study Geomatics.

OSGeo Journal:

When and how did you get involved

with open source GIS and QGIS in particular?

Anita:

We got introduced to [PostGIS](#) and [UMN Mapserver](#) during a university lab. I got involved with QGIS the following summer when I did an internship where I was asked to analyze the performance of an algorithm matching GPS points on road segments. QGIS was picked for the project since it is open source and tasks in QGIS can be automated with Python.

OSGeo Journal:

How have you been involved in the QGIS project in the past, and what is your current role?

Anita:

I'm mostly doing community work such as answering questions on forums and [gis.stackexchange](#). I'm also blogging and speaking about QGIS at conferences such as AGIT or FOSSGIS where I've presented both general QGIS news and my own plugins (for example: Time Manager).

OSGeo Journal:

How do you use QGIS in your own work?

Anita:

I use QGIS (together with PostGIS) a lot to analyze and visualize mobility data such as GPS tracks of vehicles or

persons. QGIS serves me as a tool to explore and validate data as well as to create strong visuals that present project results to the general public.

OSGeo Journal:

What plug-ins are you currently developing for QGIS?

Anita:

Currently, I'm maintaining Time Manager and pgRouting Layer plugins.

Time Manager enables you to easily browse through spatio-temporal data and create animations. (You can see some examples on <http://www.youtube.com/playlist?list=PLF97A9E490FB2563D>)

pgRouting Layer is a front end which makes calculating routes with pgRouting for PostGIS much easier because you don't have to remember the complex syntax anymore.

OSGeo Journal:

What are your favorite QGIS features?

Anita:

QGIS reads almost any data format known to the GIS world and provides the tools to both visualize and analyze spatial data in a very straight-forward way. The community is great (new features are added on a weekly basis)

and it's hard to pick favorites. I love the rule-based rendering because it's powerful and flexible. I also enjoy QGIS Server for publishing my projects as a web service. Sextante integration is definitely on my watch list as one of the great recent additions.

OSGeo Journal:

What does QGIS not do that you wish it could do?

Anita:

Highest on my wish list would be data-defined symbology. That means that for example the color of features in a layer should be defined by a color attribute.

OSGeo Journal:

Why is the OSGeo important to you, and what role do you see for the organization moving into the future?

Anita:

For me, OSGeo provides a platform that enables open source projects to present themselves in a more professional manner. Organizing events and spreading the word is central to both acceptance and success of open source alternatives.

Open Source Geospatial Software Case Study

Converting Hard Copy Engineering Records for Sanitary Sewer Networks to GIS Data Using OpenJUMP

By Landon Blake, Neal Colwell, and Julian Padilla

Project Background

KSN is a civil engineering and land surveying company based in Stockton, California, USA. For several decades KSN has been serving a number of small sanitary sewer utility districts in the California Central Valley and Sierra Nevada Foothills. Historically, the engineering data for each district was stored on hard copy basemaps, field



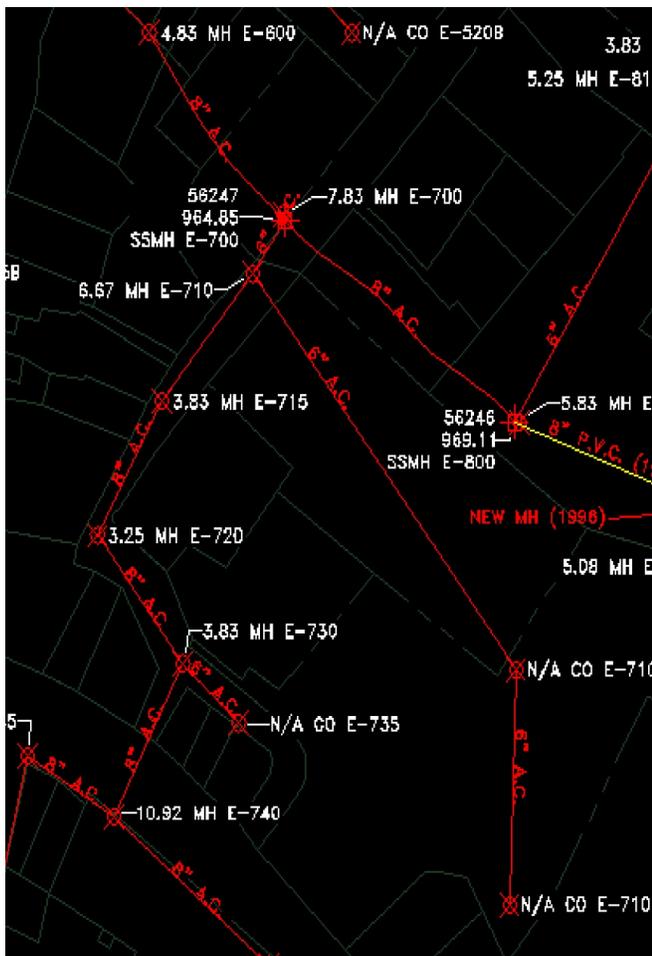
Sewer Network Hardcopy Basemap

notes, inspection reports and plan or profile sheets. In the past decade KSN began to help its clients convert some of these paper records to computer aided drafting (CAD) drawing files.

One limitation of CAD files is the difficulty in associating attribute data with feature geometry. (For example: Associating pipe size, pipe material, and pipe flowline elevations or inverts with a line geometry representing a

sewer pipe.) During the conversion from hardcopy records to CAD drawings, some of the attribute data can be annotated in the CAD drawings as text labels. However, there is a practical limit to the number of attributes that can be annotated in a CAD drawing in this manner. It is also difficult to perform modeling and analysis in CAD using these annotations.

In recent months KSN began a series of projects to convert the hardcopy



Sewer Network CAD Basemap Drawing File

engineering records and CAD files to GIS. A main goal of these conversion projects is to support modeling and analysis of the sewer networks using the GIS data produced as part of the conversion. The conversion projects are being performed with the help of open source geospatial software.

The conversion projects are being performed with the help of open source geospatial software.

OpenJUMP

The main tool being used as part of the conversion projects is [OpenJUMP](#). OpenJUMP is an open source desktop GIS program written in the Java programming language. OpenJUMP excels at the creation and editing of vector GIS data. It also features a pluggable architecture which allows for easy custom tool development and integration.

Why Open Source?

Why did KSN choose an open source desktop GIS for these projects? There were a number of reasons why OpenJUMP made sense for these projects. They included:

1) *There was no significant upfront*

investment required on the part of KSN or the sanitary districts to acquire proprietary GIS software licenses.

2) OpenJUMP is easy to learn and use, requiring little investment in staff training.

3) OpenJUMP plug-ins automate many repetitive and monotonous data entry and creation tasks.

4) OpenJUMP's pluggable architecture facilitate the development of network topology and network analysis tools that can be used with the data produced as part of the projects.

The Project Team

The conversion projects were managed and executed by three (3) key staff members at KSN.

Neal Colwell is a licensed civil engineer and project engineer for KSN. Neal has many years of sewer design, modeling, and analysis experience. He is also an experience consumer of GIS data products and immediately saw a need to convert the hardcopy records and CAD drawings for the small sanitary districts to a GIS format. Neal Colwell is managing the conversion projects and will ultimately use the GIS data produced by the projects for sewer

network modeling and design on behalf of the KSN clients.

Landon Blake is a licensed land surveyor and project manager for KSN. As an advocate of open source geospatial software, Landon continually seeks a wider adoption of GIS by engineers and surveyors. He is always looking for opportunities to apply open source geospatial software tools to engineering and surveying problems, including those tackled by KSN staff. Landon is also a volunteer administrator and programmer for OpenJUMP. Landon supervises the data conversion work on the projects and is directly responsible for the creation of the network topology and network analysis plug-ins for OpenJUMP.

Julian Padilla is an intern at KSN and an engineering student at [Delta College](#) in Stockton, California. Julian had previous experience using OpenJUMP during a high school community service project for the California Land Surveyors Association Central Valley Chapter. This experience made him the ideal person to perform data conversion work on the projects.

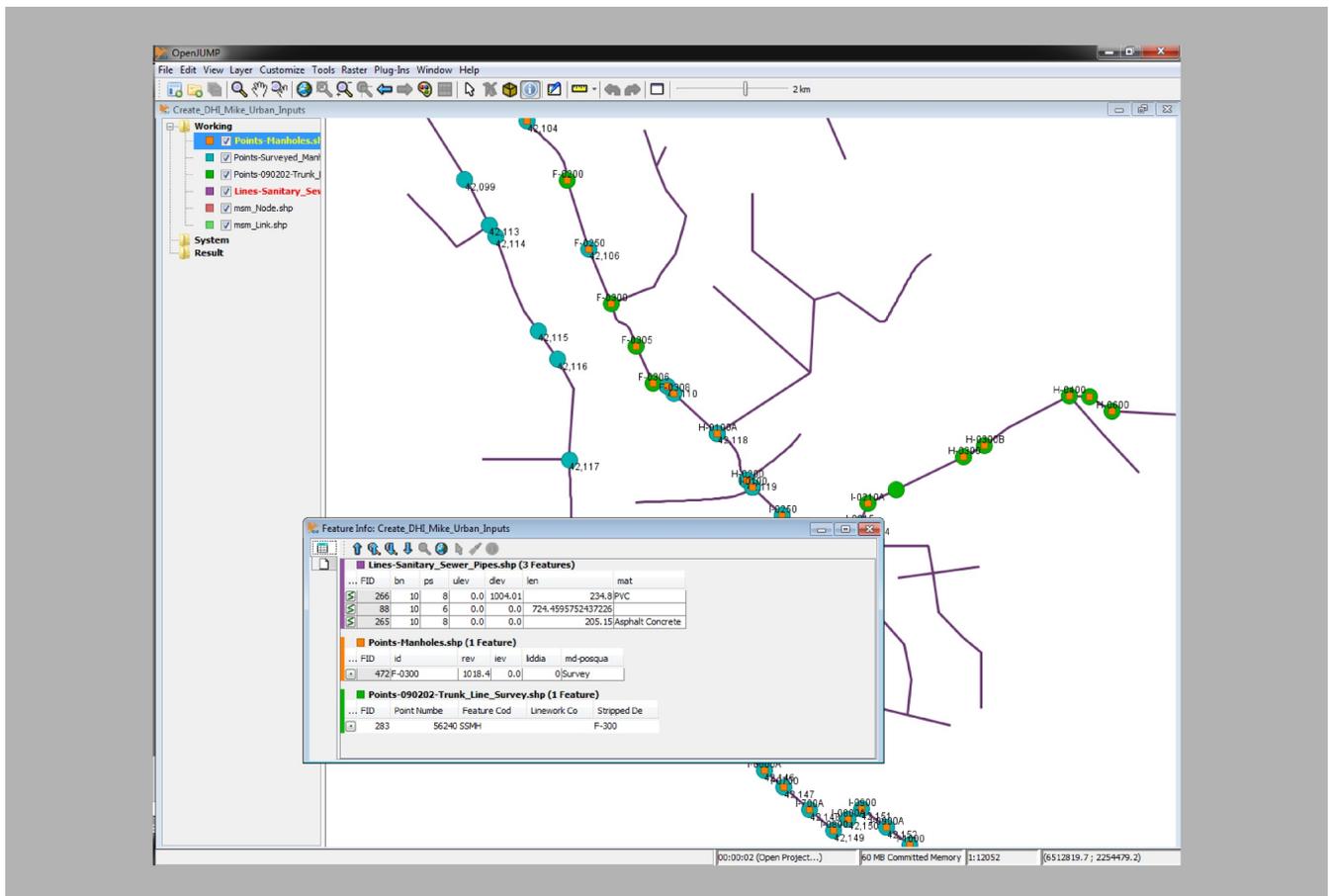
Project Execution

Two (2) different approaches to project execution are necessary. The first

approach is used for sanitary sewer districts that have CAD drawings representing all or most of their sewer networks. The second approach is used for sanitary sewer districts that still have hardcopy engineering records and little or no CAD drawings.

In the first approach the sanitary sewer network is divided into smaller networks called “branch networks”. The CAD drawing entities representing sewer pipes are segregated by pipe size and branch network. A custom AutoLISP

routine developed by Landon is used to export the pipe geometry into WKT format. OpenJUMP is then used to import the WKT pipe data. Once in OpenJUMP the pipe data is attributed from the CAD drawing file text labels and other hard copy records. Surveyed locations of manhole lids are then imported into OpenJUMP using a plug-in developed by Landon. Once in OpenJUMP, the survey data for the manholes is used to create manhole observation features. Manhole features are built from these observations.



Editing Sewer Network GIS Data in OpenJUMP

Manhole measure downs or dips taken by KSN survey field crews are then used to create pipe invert elevation attributes and manhole invert elevation attributes.

Feature level metadata, feature edit history, network topology, and spatial relationships are created and maintained for pipes and manholes during the projects.

In the second approach hardcopy basemaps and other hard copy records are used to draw pipe and manhole features. Georeferenced county GIS parcel data and aerial photography is used as a background to aid in the drawing of the sewer network in OpenJUMP. Once the vector geometry is created, attributes are added to the pipes and manholes using information in the hard copy records.

Feature level metadata, feature edit history, network topology, and spatial relationships are created and maintained for pipes and manholes during the projects. Network topology is stored in node and connector tables. The topology tables also store the

branch identifier and position within the branch for all manholes and pipes.

Network Analysis and Topology Tools

Landon Blake is currently developing OpenJUMP plug-ins to create, manage, and analyze network topology data. Although these plug-ins will be initially used on these projects for sewer network topology data, they will be applicable to networks of all types. The source code for the plug-ins will be released under the [GPL](#) through the [SurveyOS Project](#).

Although these plug-ins will be initially used on these projects for sewer network topology data, they will be applicable to networks of all types.

Future Tasks and Opportunities

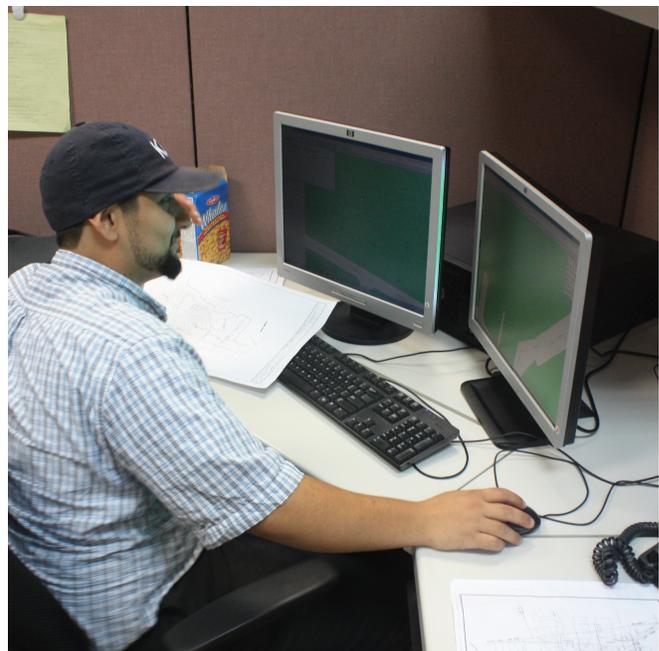
Once the initial phase of the conversion projects are complete, the KSN project team will turn its attention to additional project tasks and opportunities. These tasks and opportunities include the following:

- 1) *Field surveys to collect high-quality survey grade positions on manholes in the sewer networks that have not yet been surveyed.*
- 2) *Development of a GIS maintenance and management plan for each sanitary sewer network.*
- 3) *Development of CAD data and reporting tools for OpenJUMP that can be applied to the sanitary sewer network GIS data.*
- 4) *Sanitary sewer network basemap production using OpenJUMP and Inkscape.*
- 5) *Improvement of sanitary sewer network operation using GIS tools.*
- 6) *Development of a network query language plug-in for OpenJUMP.*

Conclusion

The sanitary sewer network conversion projects at KSN have definitely proven the concept of using open source geospatial tools in an engineering and surveying environment. OpenJUMP's ease of use and modular architecture has made it an excellent fit for KSN's GIS needs on these projects. The KSN project team looks forward to further

enhancing OpenJUMP's capabilities through their own development efforts, to sharing these improvements with the larger open source community, and to finding other applications of open source geospatial software that can benefit KSN's long time clients.



Engineering Intern Julian Padilla works to create GIS data from hard copy sewer network basemap sheets.

Sharing GIS Data Models

By Landon Blake

Introduction

In the open source software development model benefits are gained through the sharing and collaborative development of software source code. Oftentimes the sum of the benefits realized under this cooperative approach to software development is greater than the benefits resulting from the work of the individuals. Similar benefits can be realized from the collaborative design and sharing of GIS data models.

This is the first in a series of articles about the design and sharing of GIS data models. In this first article we will:

- 1) Define a GIS data model.*
- 2) Describe the basic elements of a GIS data model.*
- 3) Provide an example of one way to document and share information about a simple feature class.*
- 4) Discuss two (2) example diagrams*

used to document the simple feature types that make up a GIS data model.

The contents of this article are based, in part, on work I've completed while designing the sanitary sewer network GIS data model as part of my volunteer work for the [Utility Working Group of CCVGPG](#) (California Central Valley Geospatial Professionals Group). We will begin the meat of this article with a definition of a GIS data model.

Simple Definition of a GIS Data Model

Here is a simple definition of a GIS data model: A GIS data model is a blueprint for the implementation of a geographic information system (GIS).

Benefits can be realized from the collaborative design and sharing of GIS data models.

We might further expand on our definition by adding this statement: A GIS data model is typically focused on a single theme or problem domain.

Examples of problem domains include land parcels, addressing, forestry, geology, and public health.

The typical GIS data model includes a description of:

- 1) *Structural elements of the data model.*
- 2) *Policies or procedures that are used to create, delete, and modify these structural elements.*

Let's briefly consider a list of the structural elements that could be included in a GIS data model:

- 1) *Simple Feature Types*
- 2) *Topology*
- 3) *Non Spatial Entities*
- 4) *Spatial Relationships*
- 5) *Non-Spatial Relationships*
- 6) *Layer Groups*

Future articles in this series will further examine each of these structural elements of a GIS data model. In the first article we will consider the first item in the list, simple feature types, in more detail.

Simple Feature Classes

A simple feature is composed of a two

elements. The first element is a geometry representing the feature's shape and location. The second element is a set of attributes that capture non-spatial information about the feature. For example: You could represent a road feature using a linear geometry, an attribute that stores the road name, and an attribute that stores the road type.

A GIS data model is a blueprint for the implementation of a geographic information system (GIS). A GIS data model is typically focused on a single theme or problem domain.

The type of geometry used to represent a feature, and the set of attributes common to all features of a single type, are typically defined in a feature schema or feature type. These simple feature types are the basic building block that forms the foundation of a GIS data model. All other data model elements related (directly or indirectly) to the simple feature types defined in the GIS data model.

Structural Elements of a GIS Data Model

- Simple Feature Types
- Topology
- Non Spatial Entities
- Spatial Relationships
- Layer Groups

Documenting Simple Feature Types

How can simple feature types be documented in a GIS data model? A number of diagrams may be needed to document the simple feature types in a GIS data model these diagrams include the following:

- 1) An entity relationship diagram showing the relationship between simple feature types and other structural elements of the GIS data model.*
- 2) A simple feature type attribute diagram, which provides information on the name, data type, and description of each attribute that stores non-spatial data about features of the feature type.*
- 3) Policy diagrams that explain how the features of the feature type are created, destroyed, and modified.*

4) Attribute domain diagrams that list valid values for attributes of the feature type.

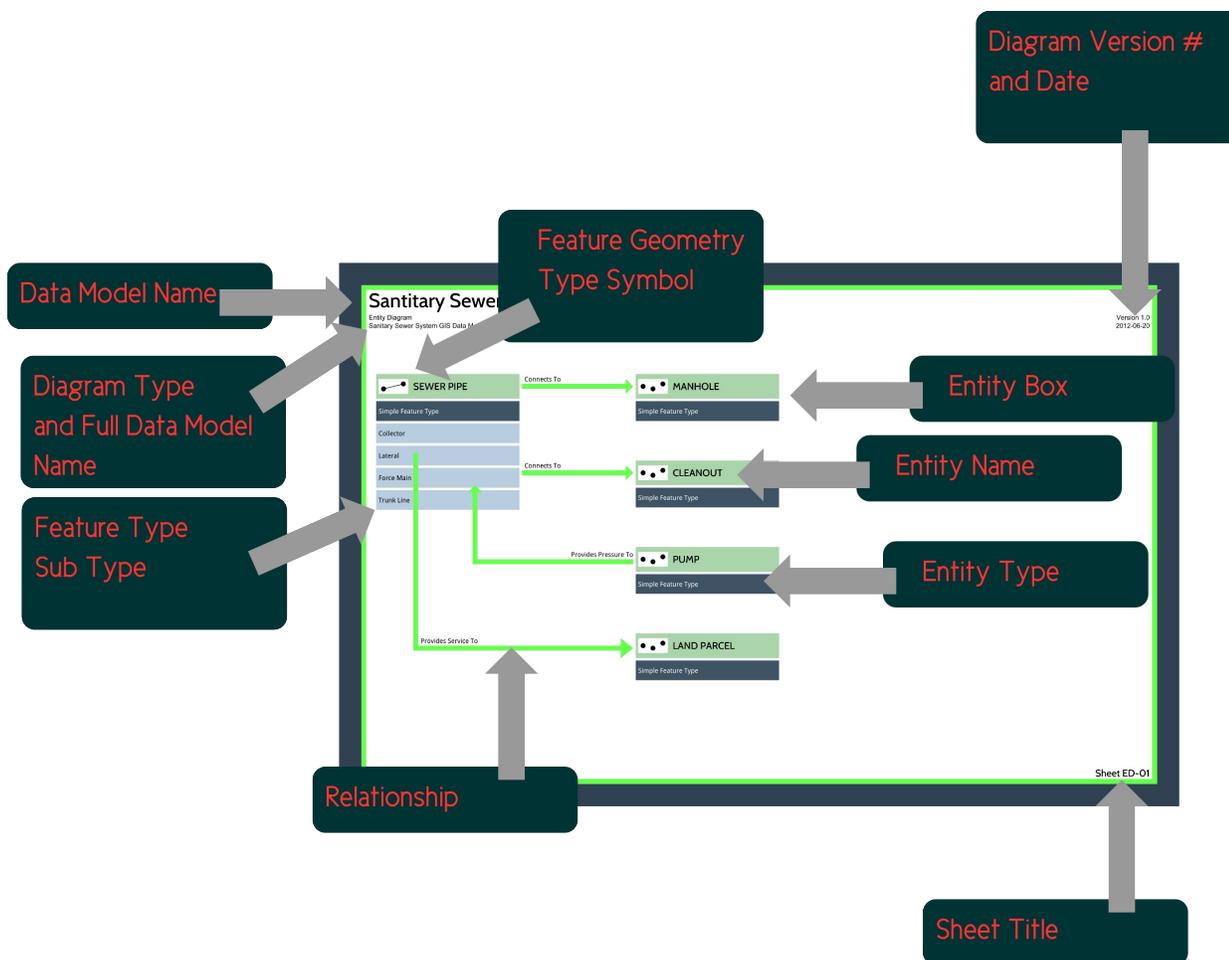
5) Sub type diagrams that list valid attribute values for all features of a feature type that belong to a more specific sub type. (For example: You might require that all road features that represent state highways have a required set of attribute values.)

Examples

We provide examples of the first two (2) types of diagrams in this article. The examples are taken from the sanitary sewer network GIS data model being developed by the CCVGPG working group. Future articles in this series will provide examples of the other the

Simple Feature Type Diagram Attribute

- Full Attribute Name
- Attribute Name Abbreviation
- Required Value?
- Data Type
- Unique?
- Units
- Domain
- Default Value



Annotated Entity Diagram

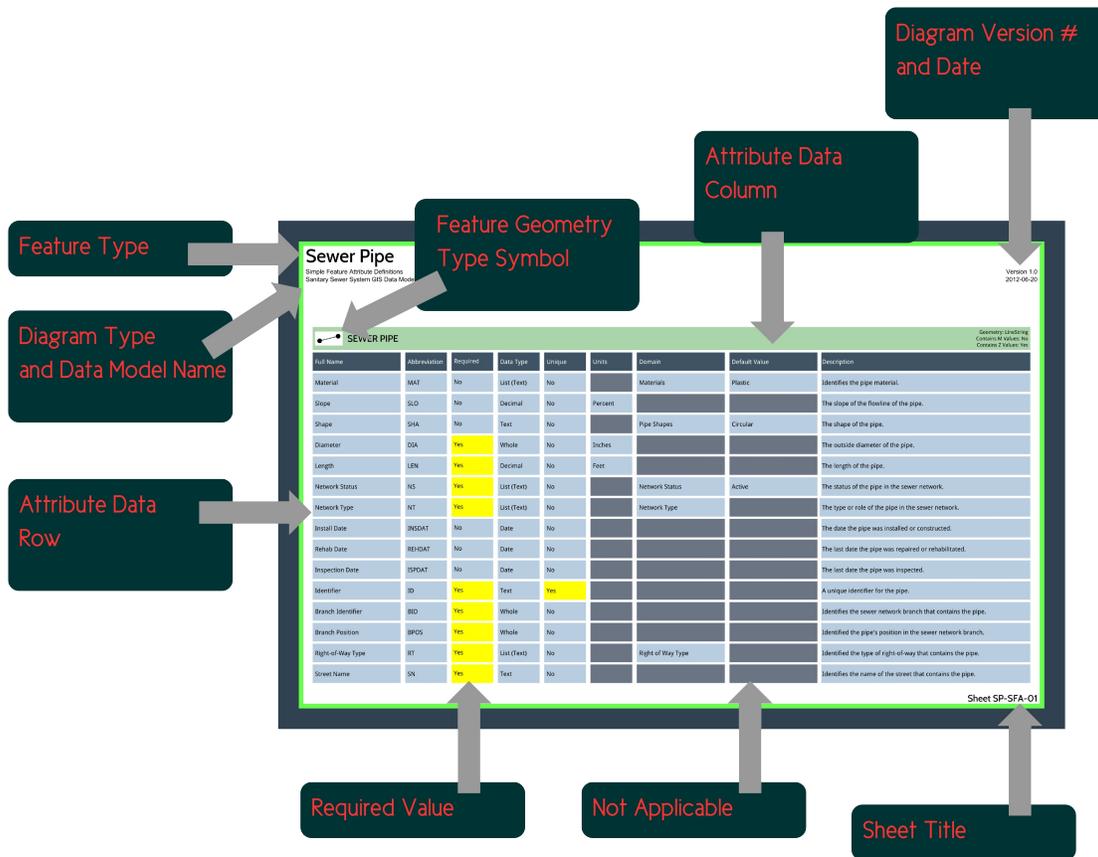
diagram types we listed in this article.

The Entity Relationship Diagram

The entity relationship diagram shows the simple feature types and non-spatial entities that make up the GIS data model as well as the relationships between them. It provides a simple “high-level” view of the GIS data model.

The entity relationship diagram has two (2) main components. The first component is an entity box. The second component is the relationship arrow. We will briefly describe each of these.

The entity box composed of three (3) types of smaller boxes or bars. The top bar shows the name of the entity, and if the entity is a simple feature type, the geometry type that



Annotated Simple Feature Type Attribute Diagram

is used to represent features of the type. The second bar shows the type of entity described by the box. This could be a simple feature type or a non-spatial entity.

The third bar shows all of the sub types of a simple feature type or a non-spatial entity.

The relationship arrow has only two (2) parts. The first is an arrow that shows the two elements of the GIS data model

that are participating in the relationship. This could be a simple feature type, non-spatial entity, or a subtype of either. The second part of the relationship arrow is the name of the relationship.

The Simple Feature Type Attribute Diagram

The simple feature type attribute diagram describes the attributes of a

simple feature type in a standard way. A table is used to represent the simple feature type in the diagram. The first row of the table shows the geometry type of simple feature type and the feature name. It also indicates if the feature geometry contains elevation values or measure values (route stationing).

The remainder of the table contains data about the simple feature type attribute schema. For each attribute in the schema the following information is displayed:

- 1) *The full name of the attribute.*
- 2) *The attribute name abbreviation.*
- 3) *If the attribute value is required.*
- 4) *The data type of the attribute value.*
- 5) *If the attribute value must be unique among features of the simple feature type.*
- 6) *The units of the attribute value.*
- 7) *The attribute value domain, if applicable. (A domain is a limited list of acceptable values.)*
- 8) *The default value for the attribute.*
- 9) *A description of the attribute.*

Conclusion

In the next volume of the OSGeo Journal we will take a look at some of the other diagram types listed above. We'll also look at how the simple feature type attribute diagram can be represented as **JSON** and exchanged between GIS software for the automatic creation of GIS data model layers.

The SurveyOS KML Toolkit: Starting With Simple Placemarks

By Landon Blake

Introduction

This is the first installment of a new column in the OSGeo Journal. This column will follow the development of a brand new open source geospatial software project: The SurveyOS KML Toolkit. The software project will develop a library and front-end GUI application to create and manage [KML](#) entities.

This column has 3 goals:

- 1) Teach the basic concepts of Ruby object-oriented programming.*
- 2) Teach the concepts of KML.*
- 3) Develop an open source KML toolkit suitable for a candidate as an OSGeo Labs Project.*

Do You Want To Learn More?

You can learn more about Ruby programming here:
<http://www.ruby-lang.org/en/>

You can learn more about object-oriented programming here:
<http://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep>

I've been writing open source software in Java for several years, but I'm new to Ruby programming. I'm also not an expert at Google's KML. In this column I will be learning along with my readers, and I look forward to getting constructive feedback from readers with experience in these two (2) areas of technology.

The first component of the library will be a Ruby programming library. The second component will be a front-end GUI program.

Before we discuss the basic concepts and goals behind the SurveyOS KML Toolkit, I thought it would be helpful to talk a little bit about Ruby, KML, and the SurveyOS Project. This discussion will provide some helpful background for the rest of the article.

You should be familiar with object-oriented programming basics to benefit from this article. A bit of knowledge about Ruby programming is also helpful, but isn't critical if you've programmed in other object-oriented programming languages before.

A Little Bit About the Ruby Programming Language

Ruby was developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan. It is a dynamic, reflective, and interpreted language. It supports different programming styles, such as object-oriented programming and functional programming.

There are a number of interpreters that can execute Ruby code, but YARV is

bundled in the standard 1.9 distribution of Ruby.

Ruby is used as a scripting language in one of my favorite proprietary programs, SketchUp.

A Little Bit About KML

KML (Keyhole Markup Language) is an XML language that can be used to represent and visualize geospatial information. It was initially developed for use in the Keyhole Earth Viewer, but has been adopted by the OGC as a standard.

A Little Bit About the SurveyOS Project

The SurveyOS Project is focused on the creation and management of open source software to increase the ability of land surveyors to create and work with GIS data. The SurveyOS Project includes a number of sub-projects. The sub-projects include software libraries and applications written in Java, Python, Visual Basic .Net, and AutoLISP.

The Basic Concepts and Goals of the KML Toolkit

Before we begin to dissect the first bit of source code for the KML Toolkit, we'll

Do You Want To Learn Participate?

The source code for the SurveyOS Project KML Toolkit can be found here:
<http://surveyos.svn.sourceforge.net/viewvc/surveyos/ruby/kmltools/>

Contact Landon Blake by e-mail at sunburned.surveyor@gmail.com if you have questions, want to contribute code, documentation, or serve as a beta tester.

discuss briefly the basic concepts behind the toolkit. We'll also talk about some of the design goals for the toolkit.

The toolkit will be divided into two (2) main components. The first component will be a Ruby programming library that can be easily integrated into other Ruby applications. The second component will be a front-end GUI program that exposes the functionality of the library.

The typical work-flow of a user or client application with the KML Toolkit will involve these three (3) steps:

1) Build a collection of KML entities.

KML entities will be added to the collection using one (1) of two (2) methods.

In the first method the user will use the toolkit to import existing spatial data which will be converted to KML entities. (For example: Import of an existing ESRI shapefile or text delimited file that stores data about point features that are then converted to KML placemark entities.)

In the second method KML entities will be automatically created based on a set of input criteria. (For example: Creating KML placemark entities on a regular

grid, at every intersection of linear features, or at regular intervals along a linear feature.)

2) Organize and manage the collection of KML entities.

Once KML entities have been added to a collection, they can be organized. Two (2) primary ways to organize the data will be available. One way will be with layers. A layer holds KML entities of the same type and with the same attributes. (For example: One layer might hold placemarks for industrial buildings, while another holds placemarks for commercial buildings.)

A second way the collection of KML entities can be organized is with groups. A group can be made up of KML entities of different types that share some high level relationship. (For example: You might group paths and placemarks representing a railroad network.)

The user will also be able to query and manage selections of KML entities based on their identity, properties, and descriptions. (For example: Select all of the paths of type "highway" that are over 10 miles in length.)

3) Export KML documents.

After a collection of KML entities is complete, the user (or client application) can export actual KML documents based on the collection of KML entities. (For example: This would allow the user to export KML documents for different audiences based on the same collection of KML entities.) The user will be able to style entities in the exported KML document using style templates. Style templates can be applied to KML layers or groups.

Design Goals

The SurveyOS KML Toolkit has the following high-level design goals:

- 1) Support of a plug-in framework for easy extension by third party programmers.*
- 2) Support for an undo/redo framework.*
- 3) Support for clean separation between GUI and core program code.*

Getting Started with the SimplePlacemark Class

In my object-oriented programming projects I find it helpful to start the design of a library or program with the simple core data objects. I find that

most of my software projects will only have a handful of core data objects. (Many of my projects have only a single core data object.)

After the core data objects are designed, I work on adding a “program” structure to my software project. This program structure usually hosts a structure to contain instances of the core data object and a framework for tools that can create, manipulate and manage these core data objects.

It is helpful to start the design of a library or program with the simple core data objects.

I started the design of the SurveyOS KML Toolkit with a Ruby class representing a single core data object. This class is used to represent the simplest sort of placemark KML entity. Before we look at the design of the class itself, let’s take a look at how such a placemark would actually look in a KML document. You can see this in Source Code Listing #1 shown at the end of this article.

We can see that our class needs to store a name, description, and coordinate for the placemark. That’s

exactly what we do in the SimplePlacemark class.

Source Code Listing #2 contains the source code for the current version of the SimplePlacemark class. On the next page is a graphical overview of the SimplePlacemark class member variables and methods.

All of the member variables, or data for our SimplePlacemark class are simple Ruby primitives, except for the Coordinate class. The Coordinate class data is defined entirely with Ruby data primitives. I call this type of class a terminal data class. (A terminal data class doesn't reference any external class definitions, only data primitives.) They are the simplest type of class. The Coordinate class simply bundles up three floating point number values that represent a latitude, longitude and elevation.

Whenever I have a set of simple data values that will often be passed around my class as a set, I consider making a terminal data class to clarify my source code. The Coordinate class is just this type of class.

You can see the methods of our class can also be organized into three (3) groups:

1) Accessor methods that allow access to the member variables.

2) Methods to compare the equality of member data.

3) Common utility methods that should be implemented on most core data classes for a program, including a clone method and a method to represent the core data object as a string.

What's Next

In the next article we will take a look at the implementation for a couple of the methods of the SimplePlacemarkClass. We will also look at the unit test we designed for the class. Then we can start poking around the collection class that will hold our SimplePlacemark objects and that will form the core of our Program class.

Conclusion

In this article we've accomplished the following tasks:

1) We've looked at the concepts and design goals for our toolkit.

2) We've examined the design of our first core data object, which is defined

SimplePlacemark Class

Member Data		
Variable Name	Data Type	Description
@name	String	Stores the name of the placemark.
@description	String	Stores the description of the placemark.
@latitude	Float	Stores the latitude of the placemark location.
@longitude	Float	Stores the longitude of the placemark location.
@coordinate	Coordinate	Stores the location of the placemark as a Coordinate.

Accessor Methods	
Method Name	Return Value Data Type
get_name	Float
get_description	String
get_latitude	Float
get_longitude	Float
get_coordinate	Coordinate

Comparison Methods	
Method Name	Return Value Data Type
has_same_name	Boolean
has_same_description	Boolean
has_same_location	Boolean

Utility Methods	
Method Name	Return Value Data Type
clone	Simple Placemark
to_string	String

by the SimplePlacemark class.

There are other classes (including a unit test for the SimplePlacemark class) in the SurveyOS SVN repository folder for this software project. If you are interested in what you've read in this article, you might look further at the source code found there.

Source Code Listing #1

```
<Placemark>
  <name>Landon's House</name>
  <description>A placemark representing Landon's house. </description>
  <Point>
    <coordinates>-121.10233356, 37.9255487, 0</coordinates>
  </Point>
</Placemark>
```

Source Code Listing #2

```
# Represents a KML placemark. This simple version of a placemark stores and allows
# access to the placemark name, the latitude and longitude of the placemark, and
# a simple placemark description with no embedded HTML. This class is immutable.
class SimplePlacemark
  include KMLEntity

  # Creates a new placemark.
  #
  # name          = The name of the placemark as a string.
  # latitude      = The latitude of the placemark in decimal degrees as a double.
  # longitude     = The longitude of the placemark in decimal degrees as a double.
  # description   = The description of the placemark as a string.
  def initialize( name, coordinate, description)
    @name = name
    @coordinate = coordinate
    @latitude = coordinate.get_latitude();
    @longitude = coordinate.get_longitude();
    @description = description
  end # End constructor.

  def get_name()
    return @name
  end # End method.

  def get_description()
    return @description
  end # End method.
```

```
def get_coordinate()
  return @coordinate
end

def get_latitude()
  return @latitude
end

def get_longitude()
  return @longitude
end

def to_string()

  # Convert the latitude and longitude to strings.
  latitudeAsString = @latitude.to_s
  longitudeAsString = @longitude.to_s

  placemarkAsString = "Placemark{Name: \"#{@name}\", \" + "Latitude: " +
latitudeAsString + ", \" + "Longitude: " + longitudeAsString + ", \" + "Description:
\"#{@description}\"}"}"
  return placemarkAsString
end # End method.

def has_same_name(placemark)
  if
    @name == placemark.get_name()
    return true
  else
    return false
  end # End if.
end # End method.

def has_same_description(placemark)
  if
    @description == placemark.get_description()
    return true
  else
    return false
  end # End if.
end # End method.
```

```
def has_same_location(placemark)
  equality_counter = 0

  if
    @coordinate.get_latitude() == placemark.get_latitude()
    equality_counter = 1
  end # End if.

  if
    @coordinate.get_longitude() == placemark.get_longitude()
    equality_counter = equality_counter + 1
  end # End if.

  if
    equality_counter == 2
    return true
  else
    return false
  end #End if.

end # End method.

def clone()
  clone = SimplePlacemark.new(@name, @coordinate, @description)
  return clone
end # End method.

end # End class.
```

Editor's Footnote: Do We Need a Journal?

By Landon Blake

When I was first seeking contributions for this annual report from OSGeo Chapters and Software Projects I received an thought-provoking e-mail from Cameron Shorter about the return on investment realized by the busy programmers who would need to stop coding to help write the annual report item for their software project. At its essence, I think Cameron's e-mail was asking if the effort was worth it.

Cameron's e-mail was excellent. I wanted to conclude Volume 11 of the OSGeo Journal with a short editor's footnote that responded to Cameron's intelligent questions, and spoke to some of the broader questions I think his e-mail raised.

Before we answer the larger question that deals with the importance of having a journal for our organization, let me specifically address questions about our annual report.

Who is the target audience for the annual report?

I segregate our target audience into two (2) groups. In the first group are OSGeo members or people actively involved in the open source geospatial software development community. For this group, the annual report provides an opportunity to evaluate the efforts of their peers. I hope the annual report provides these readers with an opportunity to be inspired by what others are doing.

The second group is people that aren't actively involved in OSGeo or open source geospatial software, but are starting to explore those areas. They might be regular users of proprietary geospatial software, and they've seen one or two open source tools that have sparked their interest.

What will attract readers to the annual report?

The main thing I can do as the editor of (and you can do as a contributor to) the annual report, is ensure we have content that is enjoyable to read. The quality of the annual report items and

the level of participation by OSGeo chapters, sponsors, and software projects are important factors.

I'm taking one (1) important step towards improving the quality of our annual report items currently. I'm in the midst of writing two (2) guides that will help authors contribute better annual reports. The first guide will be for OSGeo local chapters, while the second guide will be for OSGeo software projects.

The level of participation by chapters and software projects can be increased mainly through your own efforts. I'll be working on annual report item templates that will make this a bit easier, and the guides should help to. The journal already takes care of annual report item review, copyediting, and formatting for contributors.

I'm also striving to include at least a couple of articles not directly related to the annual report in each annual report. In this issue we had an open source geospatial software case study, an interview with a super user, a topical article about sharing GIS data models, and an article about a new software toolkit written in Ruby. These additional articles should spice up the annual report and make it more appealing.

How is the OSGeo Journal being marketed?

As the editor, my marketing efforts will focus on the following tasks:

1) Maintenance of the OSGeo Journal Blog.

2) Conversion of the journal content into a variety of formats, including HTML, EPUB, PDF and content for publication in Google Currents.

The rest of the marketing, at least for the time being, is up to you. If you have specific ideas about how to increase the number of people reading the journal, please discuss it with our volunteer team.

Do we need a journal?

I want to conclude this editor's footnote by answering this important question. It is a question I've given a lot of careful thought to. It takes a great deal of effort to write, copyedit, format, and publish the Journal. Many hours are contributed by the Journal team members.

Is this investment worth the cost?

This is still a difficult question for me to

answer. For those that are skeptical about the return on investment, I offer this thought for consideration:

How many non-profit organizations serious about accomplishing their mission don't utilize a newsletter or other publication to communicate to their supporters and to prospective supporters?

I can't think of any.

Should the OSGeo be an exception?

I'll finish this last section of the editor's footnote with a simple list of the reasons I feel it is important for us to continue producing a journal:

- 1) It provides the OSGeo with a voice and an opportunity to reach out to potential supporters and sponsors.*
- 2) It provides excellent marketing for OSGeo software projects and local chapters.*
- 3) It helps bring our diverse, global community together, by informing OSGeo members of the OSGeo activities taking place around the world.*
- 4) It provides the opportunity for software projects to write quality*

tutorials, project documentation, and case studies, which often not produced otherwise.

The Editor

Imprint

The Journal Team Editor:

Landon Blake

Section Editors:

Scott Mitchell

Reviewers:

Eli Adam

Credits:

The deegree Software Project, GRASS Software Project, GeoMoose Software Project, QGIS Software Project, California Chapter, Francophone Chapter, Greek Chapter, Italian Chapter, Korean Chapter, and pycsw Software Project all contributed content for this volume. We thank them for their support and contributions.

The OSGeo Journal is a publication of the OSGeo Foundation. The open source tools Scribus, Inkscape, Tortoise SVN, and Libre Office were used in the production of the Journal.

License and Copyright

This work is licensed under a Creative Commons Attribution-No-NoDerivs 3.0 United States License. To view a copy of this license, visit: http://creativecommons.org/licenses/by-nd/3.0/us/deed.en_US

All articles are copyrighted by the respective authors.

Web Sites

The OSGeo Journal home page can be found at: <http://www.osgeo.org/journal>

The OSGeo Journal Editor's Blog can be found

at:

<http://geojournaleditor.com/>

OSGeo Homepage: <http://www.osgeo.org/>

The editor can be reached at sunburned.surveyor@redefinedhorizons.com.

