OSGeo
Your Open Source Compass

OSGeo Journal

Volume 6 - September 2010


VISITDENVER.com
VISIT DENVER, THE CONVENTION & VISITORS BUREAU
THE MILE HIGH CITY
DENVER
FOSS4G 2011
Denver, USA
See You There!
2010
Official Visitors Guide
More information coming soon..
LIKE AN EGYPTIAN
with King Tut at the Denver Art Museum
Page 72
July 1, 2010 – January 2, 2011
"WESTERN HEMIS-FAIR"
of art, culture and music at the Biennial of the Americas
Page 76
July 1 – July 31, 2010

From the Editor...

by Tyler Mitchell

Welcome to the first edition of the OSGeo Journal for 2010! As a good kick-off to the new year this volume takes a few different perspectives on software development and design. Naturally the various issues related to typical development projects applies quite well to our open source geospatial specific interests. The articles cover a range of topics from a review of various software to a discussion of user-centered design. Along the way you'll also get to read some more technically meaty articles and some perspective pieces.

Each volume of the Journal takes several months of concerted effort by many individuals. Landon Blake played a lead editorial role in getting this volume pulled together so you can read it - thank you Landon! It's always a pleasure to have more section editors, LaTeX masters and reviewers come to help. Thank you to all the volunteers.

With our new online management system, any potential article can be submitted at anytime by simply filling in a form at http://osgeo.org/ojs. As well, over the next couple of months keep one eye open for the OSGeo 2009 Annual Report. Get your articles in soon if you have not already. Enjoy the articles!

Tyler Mitchell
Editor in Chief, OSGeo Journal
http://osgeo.org
tmitchell@osgeo.org

Contents of this volume:

Editorial 1
From the Editor... 1

News 2
Brief News from the OSGeo Community 2

Event Reports 6
CASCADOSS International Symposium and International Information Workshop 7
Summer Training Courses on FOSS4G, 2007-2009 14

Topical Interest 18
Why Every Open Source Project Needs a Good Dictator 18

Programming Tutorials 20
GPGPU With GDAL 20

Case Studies 23
gvSIG is a viable robust alternative to commercially available GIS packages 23

Project Introductions 26
GRASS Image Processing Environment 27
SEXTANTE, a Free Platform for Geospatial Analysis 32

Peer-review Papers 40
Usability Trumps Features 40

Volunteer Recognition 53
53

News

Brief News from the OSGeo Community

Compiled by Scott Mitchell

To keep abreast of OSGeo news, watch http://www.osgeo.org/news, or subscribe to its RSS feed. This report includes highlights from recent months, plus items specifically sent to the News Editor.

OSGeo Governance

Four vacant positions on the OSGeo Board of Directors were recently filled. Two new members were elected (Daniel Morissette and Tim Schaub), and two members were re-elected (Arnulf Christl and Frank Wamerdam). The new and returning directors join Ravi Kumar, Jeff McKenna, Markus Neteler, Chris Schmidt and Geoff Zeiss in making up the full Board. Board members are elected by charter members, and the tough decisions were made thanks to very high voter turnout from those members. The Board positions have a two year term, with half the positions up for election each year. The charter members of OSGeo are listed at http://www.osgeo.org/charter_members; a new charter member intake process is scheduled annually.

Conferences

FOSS4G 2009 - Sydney

The beautiful city of Sydney, Australia hosted the 2009 version of FOSS4G. This annual event is considered the premier conference for the open source geospatial community, aiming for a "full-immersion experience in established and leading edge geospatial technologies for developers, users, and people new [to the open source geospatial community]." Highlights included an array of technical workshops, updates on OSGeo projects, the "WMS Performance Shootout" (http://www.osgeo.org/news/2009/foss4g_wmsshootout), and the awarding of the 2009 Sol Katz Award to Daniel Morissette (http://www.osgeo.org/node/970).

FOSS4G 2010 - Barcelona

The FOSS4G series of conferences continued, and has just concluded, in Barcelona Spain, 6-9 September 2010 (http://2010.foss4g.org/). This year's Sol Katz winner was Professor Helena Mitasova, honouring her wide-ranging contributions to the FOSS4G community (http://www.osgeo.org/node/1069).

Local OSGeo groups: conferences and workshops

The **Québec Local Chapter** of OSGeo hosted the first Rendez-vous OSGeo Québec in Saguenay, Québec, June 15-16, 2010. This date coincided with the second anniversary of the local chapter, which was created following a meeting at the same location in June of 2008. OSGeo community members gathered to share their projects and experiences, and heard from a great lineup of invited speakers featuring many key members of the North American OSGeo community. Full details can be found at http://rendez-vous-osgeo-qc.org/2010/english/.

The **OSGeo Poland Chapter** co-organized a conference on the GIS applications of open source software. The event took place on May 13-14 in Wroclaw, in co-operation with the Institute of Geodesy and Geoinformation Science, Wroclaw University of Environmental and Life Sciences. For more info see http://www.gislab.up.wroc.pl/wogis2010/.

Ever hacked code in an Italian monastery? The **Bolsena Code Sprint 2010** was the third annual event held in a monastery overlooking Lago Bolsena. Participants report a great mix of coding progress, fantastic food, exploration of the local area, and great company. See http://www.osgeo.org/node/990 for more details, and watch for your opportunity to sign up next year.

The **German local Chapter** has announced a Call for Location (http://www.fossgis.de/konferenz/w/images/f/f5/FOSSGIS-2011_CfL.pdf) for the 2011 FOSSGIS-Konferenz (http://www.fossgis.de/konferenz). The 2010 conference was held in Osnabrück from 2-5 March, and featured a special 2-day track on Open Street Maps.

The second **Open Source GIS UK Conference** was held 21 – 22 June 2010 at CGS, University of Nottingham. Presenters included Ari Jolma, Arnulf Christl, and Tyler Mitchell, with workshops on topics including gvSIG Desktop & Mobile, OS OpenSpace and GEOSS. The conference was webcast, and recorded archives are now available at http://cgs.nottingham.ac.uk/~osgis10/os_call2010.html.

Related upcoming meetings

WebMGS2010 — the 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services — aims to provide a dedicated gathering place and open forum for researchers, professionals, government decision makers and students to exchange and discuss recent advances in web-based mapping, geospatial information services and applications. The workshop theme, Sensor Web Enablement and Geoprocessing Services, encourages diverse topics related to the new web-based development of geospatial theories, technologies and solutions. See their website http://webmgs2010.como.polimi.it for more details.

The University of Prishtina and FLOSS Kosovo are hosting the **International Software Freedom Conference of Kosova**, which will be held on September 25 and 26 at the Faculty of Electrical and Computer Engineering, University of Prishtina. Topics cover the breadth of information technology including geospatial topics. http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=8777

Google Summer of Code 2010

OSGeo participated as a mentoring organization for Google Summer of Code 2010. This program provides funding for students to work on open source projects under the support of experienced mentors. The projects participating through OSGeo include GDAL, GRASS, Mapbender, Quantum GIS, uDig, OpenJUMP, deegree, OSSIM and MapWindow.

50 good proposals were evaluated, and a very high calibre selection ended up being chosen for the 10 final positions. Students worked with their mentors to get to know their projects even better. You can view the list of 10 finalized projects at http://socghop.appspot.com/gsoc/org/home/google/gsoc2010/osgeo.

Project News

Incubation

The OSGeo Board has recently announced the approval of three projects to enter the incubation process (http://www.osgeo.org/incubator), and one graduation. Incubation is a stepping stone to becoming a full-fledged OSGeo project. Graduating incubation includes requirements for open community operation, a responsible project governance model, code provenance and license verification and general good project operation, all of which provide potential users of the software added confidence in the product's viability and safety. Geomajas, GeoServer, and MapFish commenced the incubation process, and deegree graduated.

Geomajas (http://geomajas.org/) is a free and open source GIS framework for building rich internet applications. It has sophisticated capabilities for displaying and managing geospatial information. Its modular design makes it easily extensible, while a client-server architecture guarantees endless scalability. The focus of Geomajas is to provide a platform for server-side integration of geospatial data, allowing multiple users to control and manage the data from within their own browsers. In essence, Geomajas provides a set of powerful building blocks, from which the most complex GIS applications can easily be built. A new version of the framework, 1.7.1, was released on August 11, with major improvements to the plug-in system. See http://geomajas.org/geomajas-1.7.1 for more details.

GeoServer (http://geoserver.org/) is a server written in Java that allows users to share and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards. GeoServer is the reference implementation of the Open Geospatial Consortium (OGC) Web Feature Service (WFS) and Web Coverage Service (WCS) standards, as well as a high performance certified compliant Web Map Service (WMS). GeoServer makes extensive use of the GeoTools Java library.

MapFish (http://mapfish.org/) is a flexible and complete framework for building rich web-mapping applications. It is based on the Pylons Python web framework, and emphasizes high productivity, and high-quality development. MapFish also provides a complete RIA-oriented JavaScript toolbox. The JavaScript toolbox is composed of the ExtJS, OpenLayers, GeoExt JavaScript toolkits, and specific components for interacting with MapFish web services. MapFish supports several OGC standards, including WMS, WFS, WMC, KML and GML via GeoExt and OpenLayers.

deegree (http://deegree.org/) is a Java Framework offering the main building blocks for Spatial Data Infrastructures. Its entire architecture is developed using standards of the Open Geospatial Consortium (OGC) and ISO/TC 211 (ISO Technical Committee 211 – Geographic Information/Geomatics). deegree encompasses OGC Web Services as well as Clients and security components. You are invited to share in the celebration of the project's 10th anniversary at deegree day 2010, the 16th of November, at the University of Bonn; see http://deegreeday.deegree.org/.

MapGuide

The MapGuide Open Source team has released MapGuide Open Source version 2.1.0 (http://trac.osgeo.org/mapguide/wiki/Release/2.1/Notes). Highlights of advances made with this version include many improvements in performance, scalability and stability, an improved error reporting system, raster re-projection, and CS-Map as the coordinate system library.

The project's steering committee has also announced the debut of a "Sponsorship and Donation Program" (http://mapguide.osgeo.org/sponsorship). Donations (contributions under US$500) help ensure the continued success of the project. Sponsors (contributing over US$500) receive the additional advantages of direct input to the direction of the project, priority bug report processing, and marketing benefits.

MapServer

The MapServer (http://mapserver.org/) team welcomed last holiday season and new year with the release of version 5.6, and subsequent minor releases brought them up to version 5.6.3 (as of 22 March). New features or enhancements include:

- XML Mapfile schema and XSLT
- One-pass query processing, making WFS queries on database backends much faster
- Improved control of output resolution, allowing map printing at printer resolution using a mapfile defined for screen resolution
- Labelling enhancements: ability to repeat labels along a line/multiline
- Security fixes
- Performance optimizations (one-pass image loading, optimized large shapefile handling)
- Improved OGC web services specifications support
- Improvements to MapScript.

For more details please see the full release announcement at http://lists.osgeo.org/pipermail/mapserver-users/2009-December/064001.html.

QGIS

The Quantum GIS (QGIS) project has been busy! The multi-platform, user-friendly, GPL GIS supporting a wide variety of vector, raster, and database formats (http://www.qgis.org/), released version 1.4.0, 'Enceladus' in January, and version 1.5.0 'Tethys' in July. The newest version includes around 350 bug fixes, nearly 40 new features, on top of 30 new features in version 1.4. While they claim too many changes to be able to make a short list, some highlights include a new symbology infrastructure, a field calculator, a rapidly improving printed map composer, interface improvements including a python console with syntax colouring and command history, spatial selection, render caching, custom SVG search paths, an angular measurement tool, and customizable attribute forms.

The project is also celebrating a series of recent productive developer meetings. The fourth of the year will be in Wroclaw, Poland, 12-15 November - see http://www.qgis.org/wiki/4._QGIS_Hackfest_in_Wroclaw_2010 for more details.

GDAL/OGR

The GDAL/OGR Project recently announced the release of GDAL/OGR 1.7.0. The 1.7.0 release brings many new features, including a number of new raster and vector drivers as well as many improvements to existing drivers, and limited extensions to the GDAL API. Key changes include:

- new raster drivers: BAG, EPSILON, Northwood/VerticalMapper, R, Rasterlite, SAGA GIS Binary, SRP (USRP/ASRP), EarthWatch .TIL, WKT Raster
- GDAL PCIDSK driver using the new PCIDSK SDK by default
- new vector drivers: DXF, GeoRSS, GTM, PCIDSK and VFK
- new utilities: gdaldem, gdalbuildvrt now compiled by default
- added support for Python 3.X. Compatibility with Python 2.X preserved
- removed old-generation Python bindings
- significantly improved raster drivers: GeoRaster, GeoTIFF, HFA, JPEG2000 JasPer, JPEG2000 Kakadu, NITF
- significantly improved vector drivers: CSV, KML, SQLite/SpataiLite, VRT.

Full release news is available at http://trac.osgeo.org/gdal/wiki/Release/1.7.0-News.

GRASS

GRASS development has been concentrated in three streams:

- GRASS 6.4 is the current stable release, published on September 3 (see http://grass.osgeo.org/announces/announce_grass640.html for detailed announcement). Much of the recent work has concentrated on a new wxPython-based graphical user interface, full support for Python as a scripting language, and the first full stable native Windows port. Since the 6.3.0 release in April 2008, almost 2,900 code modifications have been made, with highlighted changes including:
  - updated Lidar tools
  - new translations
  - new Python support, and
  - processing speed improvements by orders of magnitude for several modules.
- The "GRASS AddOns" mechanism allows modules to be made available outside the main distribution. This allows experimentation with new algorithms that build on GRASS capabilities without affecting the main source stability or direction. Developers can gain relatively quick and liberal access to the SVN repository for AddOns. A wiki page http://grass.osgeo.org/wiki/GRASS_AddOns shows the large and growing range of available modules.
- GRASS 7 is a major redevelopment of the source, in which backwards compatibility is not guaranteed, but conversion scripts will be provided. Many major code rewrites and redesigns are complete, underway, and in planning (http://trac.osgeo.org/grass/wiki/Grass7Planning). A lot of recent work has gone into improvements to the wxGUI and related tools, many of which will hopefully be back-ported into a future GRASS 6.4.1.

OpenLayers

A new stable release of OpenLayers, version 2.10 was announced on the 10th of September. A full list of updates can be found at http://trac.openlayers.org/wiki/Release/2.10/Notes, including new support for WMS layers with SVG as image format, for SVG enabled browsers. Previous new features, delivered by version 2.9 back in April, are listed at http://trac.openlayers.org/wiki/Release/2.9/Notes.

Event Reports

CASCADOSS International Symposium and International Information Workshop

Event Report

by M. Rusztecka, R. Wawer, E. Orlitova, A. Podolcak, and T. Steenberghen

The CASCADOSS Project's objective was to build up a critical mass of Open Source users within the GMES (Global Monitoring for Environment and Security) society (Fig. 1) that could support each other in finding open source solutions for environmental problems.

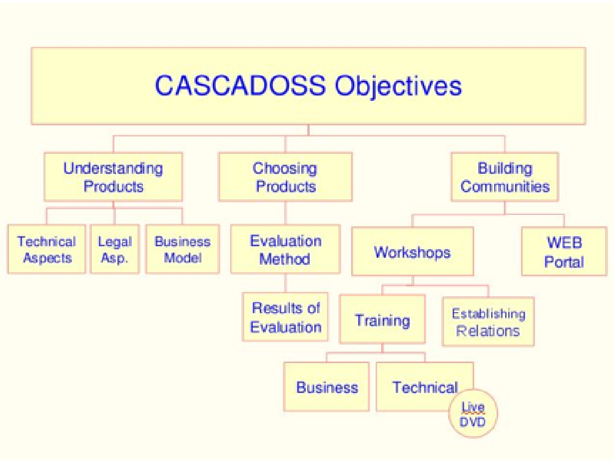


Figure 1: CASCADOSS objectives

To achieve this goal CASCADOSS set up a "transnational cascade training programme on Open Source GIS & Remote Sensing software for environmental applications". This cascading training program was divided into three phases, each aiming to refine the knowledge gained throughout the project (Fig. 2).

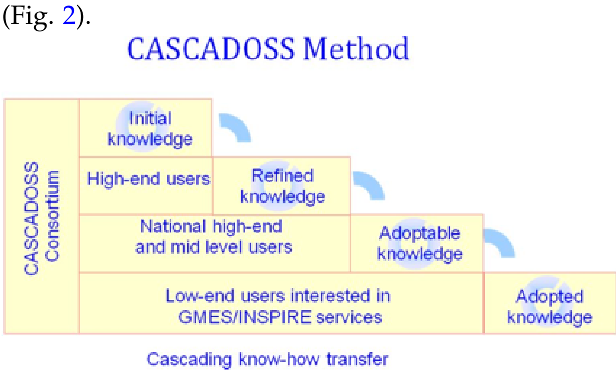


Figure 2: CASCADOSS Method

As the first step an extensive study was conducted on issues related to Open Source GIS & Remote Sensing technology:

1. A wide range of Open Source GIS & RS software projects was reviewed and evaluated. The best open Source GIS & RS projects were identified and documented,
2. A wide range of environmental applications built on top of Open Source GIS&RS Software was reviewed and evaluated. The best Open Source-based environmental applications were identified and documented.
3. The different types of business models / added value services that can be built on top of the Open Source GIS & RS technology were explored and documented.
4. Complex Open Source licensing policies were screened and translated into a comprehensive guide on Open Source legal issues.

The experience gained from this study was disseminated across Europe through the following information workshops:

- One-day International Symposium,
- 3-day International Information Workshop,
- 2-day national or regional Training Workshop.

The third and final stage was realized in the form of numerous national information workshops organized in Belgium, Poland, Slovakia, Czech Republic and Hungary. The LiveDVD of the CASCADOSS project remains a living project, hosted by Compet-Terra in Hungary: http://cascadoss.competterra.com/cascadoss.php?livedvd_en.

CASCADOSS international events

The International Information Workshop (17 to 19 June 2008) was conducted in the second phase of the CASCADOSS Project, combined with the International Symposium (16 June). The idea of the workshop was to bring together both professional developers and (potential) customers of Open Source technology and stimulate research, innovation and networking in this field. The target groups of the International Information Workshop were:

- members of organizations dealing with geo-information (GI);
- scientists;
- small and medium entrepreneurs;
- representatives of regional self-government authorities;
- staff of institutions involved in implementation of the GMES Programme, including representatives of national GMES offices;
- staff of National Mapping Authorities (NMAs);
- members of the Open-Source community.

The main purpose of the International Symposium was to present results accomplished by the CASCADOSS Project — results of the evaluation of available GIS&RS OSS solutions, and to demonstrate the most promising and useful applications.

The following subjects were also discussed:

- How can open-source software foster use of GI technologies (particularly with respect to the GMES Programme) in the public sector?
- How do existing GI standards contribute to the development of open-source applications?
- What are the current trends in this area?
- How can GIS&RS services benefit from open source solutions and stimulate development of business-related GIS&RS applications?

During the Workshop various business models or added value services that can be implemented on OSS were introduced.

Participants

Figure 3: CASCADOSS participants

There were 91 participants in the symposium and 60 in the workshop. Workshop attendance fluctuated, with roughly 20 fewer present on the third day. All of the participants were asked to conduct a short survey on their GIS/RS background, motivation for participation in the workshop, and knowledge and skills in FOSS4G — a so called "zero-measurement survey". 55 participants took part in the survey, giving answers to 5 questions. Some of the results for motivation (Fig. 4), expertise (Fig. 5) and application field (Fig. 6) are given below.

Clearly the most frequent motivation was exploration of the possibilities to use free and open source software tools in operational work.

From 40 participants polled only 17 were over the level of basic experience, while 25 (over 50%) were absolute beginners with FOSS4G, indicating however prior experience with proprietary software. In the question addressing the fields of application of FOSS4G, most responders indicated the environmental sector, dealing with scenarios of vegetation change, ecosystems, climate, water resources etc.

Quite a big group of people use FOSS4G to analyze land use, land cover changes (25) and for the purpose of web-GIS services. A number of them use FOSS4G for thematic cartography (20), spatial planning (19), environmental planning (19), environmental management (16), environmental monitoring (15) and geodetic data survey (13). The lowest number of applicants marked risk assessment and early warning assessment (10), agriculture (10), forestry (9), environmental impact assessment reporting (8), geology (6) and transport infrastructure (5).

Event agendas

The agenda of the symposium was aimed at two goals: providing general overview on the FOSS4G in context of the GMES programme and presenting in the same context the results of the CASCADOSS project: documentation and evaluation of FOSS4G software projects, analysis of FOSS4G software licenses and business models. The 3-day long workshop was focused on three main blocks: discussion, use cases and education. The discussion was addressed during the first day, starting with a general presentation of the issues, targeted for later discussion in working groups. The presentations addressed the implementation of GMES programme in member states, usage of GIS/RS in public sector (eGovernment), standardization issues (deliverables of the HUMBOLDT project), characteristic of FOSS communities and business examples related

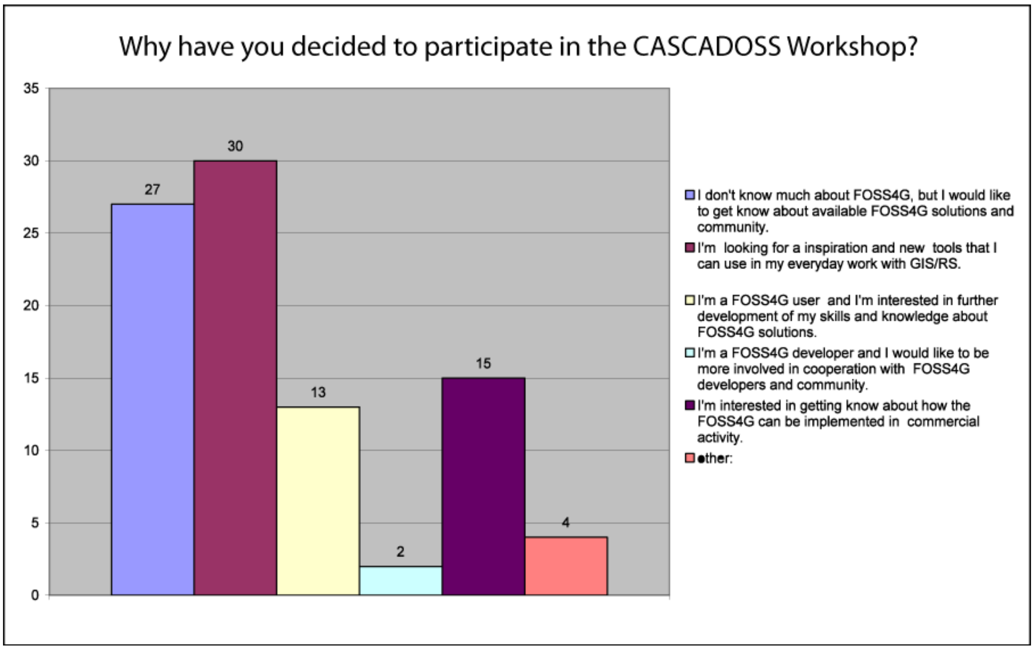


Figure 4: Reasons for participation in the CASCADOSS workshop

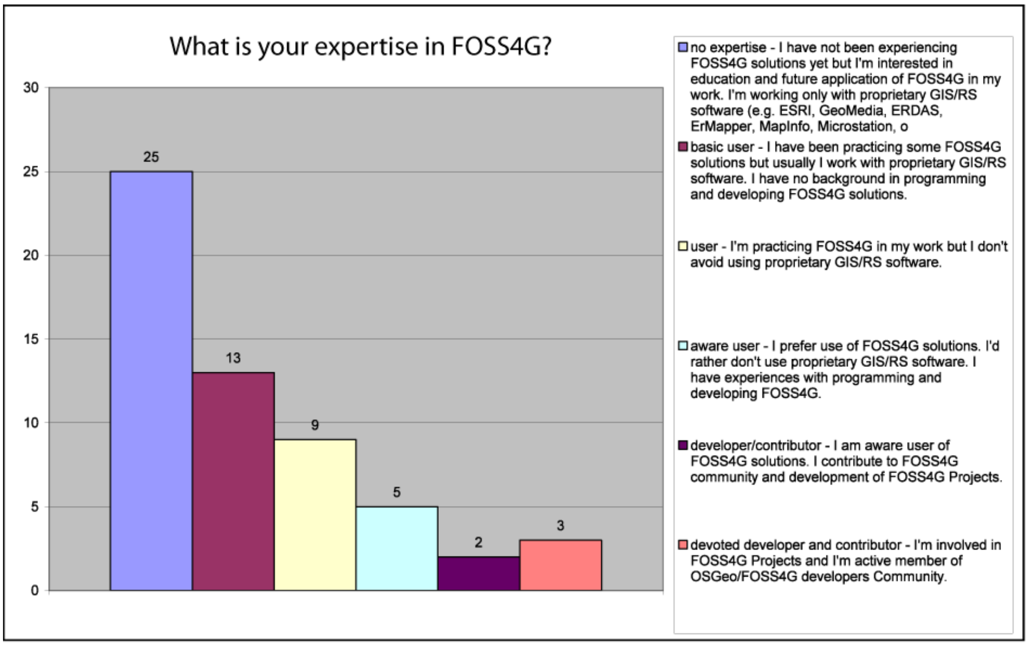


Figure 5: GIS expertise level of the CASCADOSS event participants

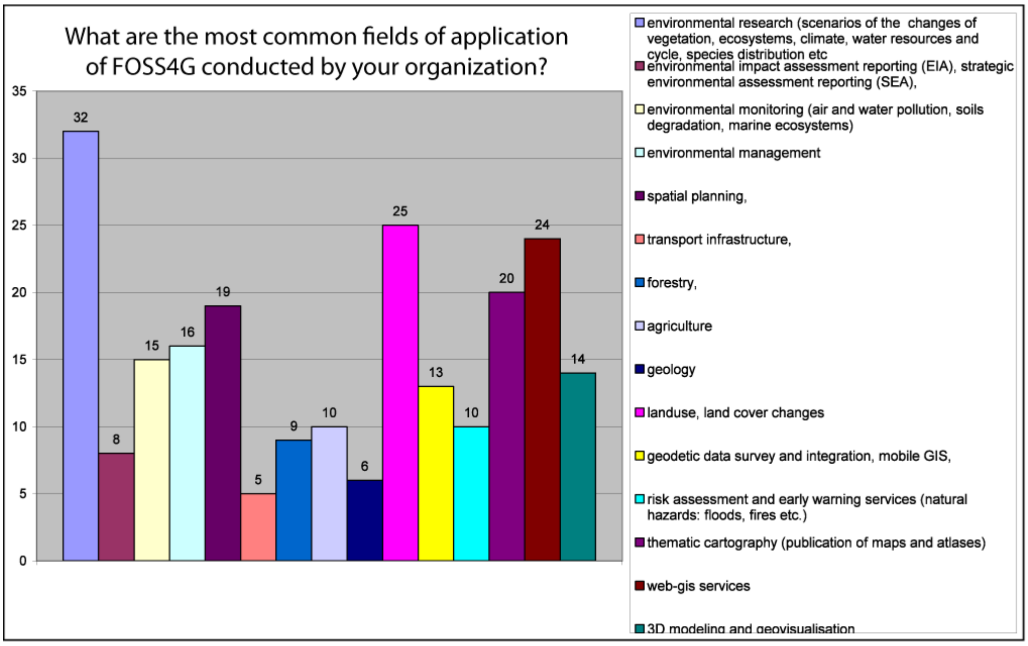


Figure 6: Most common applications of FOSS4G among CASCADOSS event participants

to the adoption of FOSS4G. The educational block was divided into three main scopes: software evaluation, GIS/RS and environmental applications. All the exercises were hands-on, utilizing CASCADOSS LiveDVD. Altogether eight successful examples of FOSS4G applications were presented during two use case sessions. More details on the event agendas can be found on the CASCADOSS website: http://www.cascadoss.eu, in the Events section.

Working Groups Discussion

Day one of the International Information Workshop was dedicated to discussion among the participants on various aspects of the use of FOSS4G solutions. In order to strengthen and facilitate the proceedings and to make them more effective, the participants had been divided into three working groups based on their professional field and area of responsibility (institution in which they work, etc.):

- WG1: FOSS4G and GI technology — for GI community, scientists and GIS practitioners;
- WG2: FOSS4G solutions in business and commercial activities — for representatives of small and medium enterprises (SMEs) and other companies and commercial entities;
- WG3: FOSS4G solutions of GMES implementation and approaches to e-government and public services — for representatives of public administration, at various levels.

Discussion covered the following issues:
FOSS4G and GI technology:

- Proprietary GI solutions versus FOSS4G: what to choose and why (a little SWOT analysis)?
- FOSS4G as a tool in GI education.
- Role of FOSS4G solutions in enhancing and developing of own IT/GI professional skills;
- What are my most common fields of application of FOSS4G solutions (sharing best practices).

FOSS4G solutions in business and commercial activities:

- Business models in practice: success stories in using FOSS4G solutions in/by the business world;
- Why SMEs seem reluctant to use FOSS4G but tend to favour commercial GI software?
- Quality and reliability of FOSS4G solutions serving as tools to provide quality service to customers.

FOSS4G solutions for GMES and approaches to e-government and public services:

- Use of FOSS4G solutions vis-à-vis legal and official procedures followed in public administration offices (for example, vis-à-vis the INSPIRE Directive implementation rules, other EU directives and standards for environmental reporting, etc.);
- Is FOSS4G implementation a challenge for public administration officials? If yes, in what sense and to which extent? How flexible would office staff be to apply new solutions in their routine professional duties?
- What legal instruments and organizational improvements can be developed to support FOSS4G implementation in the public sector?
- Which FOSS4G applications are most suitable to be used in GMES? In other words: is there a "niche" for FOSS4G in GMES (for example, for processing and analysis of GMES data)?
- Can FOSS4G applications foster and strengthen development and implementation of GMES in Europe?
- Can FOSS4G be applied to contribute to GMES mission by enhancing and improving dissemination of — and public access to — information and data on the state of environment and upcoming threats (in line with the INSPIRE Directive).

Recommendations

Results of the working group discussions are presented below in the form of lists of recommendations, agreed on between moderators and participants, regarding (within the topics specified above) proposals for:

- avoiding problems and obstacles in using FOSS4G;
- promoting use of FOSS4G;
- applying FOSS4G in the "routine", everyday practice of: (1) the GI community, (2) business sector, (3) public administration offices and GMES units/users.

Group 1: *FOSS4G and GI technology Moderators: Markus Neteler, Mateusz Łoskot* The discussion started with general questions on motivation for using FOSS4G. Among various arguments presented, the following were the strongest: ideology and FOSS4G devotion, lack of other GIS tools in work (especially in Central and Eastern Europe countries), challenge and freedom of use of such GI tools in science and academia. Consequently the SWOT (**S**trengths, **W**eaknesses, **O**pportunities, **T**hreats) analysis was conducted, the results of which are presented in Tables 1–2.

Group 2: *FOSS4G solutions in GMES and approaches to e-government and public services Moderators: Jakub Ryzenko, Ondrej Mirovski, Grzegorz Myrda, Lieven Raes, Thorsten Reitz*

Consistent with Group 1, with a public administration point of view:

1. Limited cost is good. Moreover, if you don't pay for software you can spend money on better implementation. Non-paid licenses mean unlimited number of licenses so products can be used widely and easily (e.g. by students). The potential to build communities of potential future users is easier because of non-paid licensing;
2. Strength: institution is not dependent on a particular provider of software (commercial company). So switching between software is easy, the institution is not "stuck" with a given software solution;
3. Strength: big potential in flexibility and adaptability. Can be used in specialized issues this particular institution may have;
4. OSS can expand in a bottom-up approach — number of users/institutions can become aware of possibilities created by improved software. Proprietary applications acquired in a top-bottom manner that limits their applicability. There is easier exchange of information, wider use, and flexibility of OSS;
5. From an institutional point of view, easy access by employees is better for bug-fixing and improves human relations and increases motivation of employees;
6. Lack of awareness and marketing of products, evident in the decision process (choosing particular software difficult if not widely recognized/marketed). OSS — even if similar to proprietary solutions — is weaker since less well marketed;
7. Lack of guaranteed service and customer support. For many decision makers it is not easy to find responsible individuals to make things work; unlike proprietary producers that back their products;
8. Significant effort and IT work required to get things started and the institution has to do the work themselves, rather than count on a provider;
9. With respect to GMES, discussed example of land service. It is clear that the strongest point

STRENGTHS	WEAKNESSES
- Not black-box: free and available software with possible development of functional scope. Free access to documentation and tutorials.	- Diverse point of view on "OSGeo" brand inside the community
- Flexibility of the software (FOSS4G are adjustable solutions)	- FOSS4G solutions are not so "integrated" as a package of proprietary software. It can cause problems for new users and low-end-users.
- Diversity of tools which foster development of the skills of students and researchers	- Lack of translated documentation
- Community (!!!) — OSGeo Foundation	- Lack of sample data (licensing), and information on available data resources
- Supportive of self-development (allows for networking, cooperation with other users and exchange of the knowledge and experiences. - Quick bug-fixing (action-reaction).	- Installation process of FOSS4G

Table 1: SWOT analysis for the use of FOSS4G software: Strengths and Weaknesses

OPPORTUNITIES	THREATS
- Portability — FOSS4G is portable and developed for various OS platforms (not only MS Windows)	- Domination of proprietary GI software
- PUBLIC SECTOR NEEDS — FOSS4G can be a solution for broad implementation in public sector (self-governmental units)	- Closed formats of GIS data
- INSPIRE (implementation rules of data exchange and accessibility)	- FUD effect: Fear, Uncertainty, Doubts
- DO your own business with FOSS4G — it is free and flexible so allows for creative approach in development of GI products and/or services for end-users	- Lack of education in use of GI even at a basic level — need for a broad GI education for public administration
- Make money on well prepared and designed documentation for end-users	
- FOSS4G courses and trainings	

Table 2: SWOT analysis for the use of FOSS4G software: Opportunities and Threats

is making data available to the public, not just institutions — assuming software policy allows for such dissemination;

10. The above problems can block use of OSS in GMES services, partly because of conservative institutions and FUD.

Group 3: *FOSS4G solutions in business and commercial activities Moderator: Karel Maesen*

1. Why used: price, license (no license for clients, no need for piracy of licenses), installation, operational ability to modify or extend the product, transparency.
2. Why not used: missing knowledge (awareness on using and consequences), shortage of expertise, shortages in usability and productivity, lack of simply available and customized training, packaging (download, install, run — this is not that easy as in many proprietary packages).
3. Standards: vendor lock by file format (cross-compatibility), GML data created by different vendors are not compatible. Shapefile format still works best.
4. Recommendations: EU governments communicate their experiences; funding for training, documentation; reducing liability risks: cheap indemnification insurance; reducing market fragmentation: support EU-wide market by procurement policy, since in Europe market forces seem sufficient.

The discussions and recommendations arising from them were the key achievement of the CASCADOSS International Information Workshop. We consider the workshop to have reached its goals of raising the awareness of the usefulness of FOSS4G among the end users from environmental and wider GI sectors — the potential addressees of the GMES programme.

Epilogue

The final workshop evaluation questionnaire, distributed during the workshop's last day, revealed that almost 50% of the 40 responses confirmed their willingness to use FOSS4G in daily work, 25% indicated partial interest, while one person did not intend to deal with FOSS4G at all. Through the international events in Warsaw and other dissemination activities, CASCADOSS managed to reach a wide audience of GI-users in its target geographical area

(BeNeLUx, Poland, Czech Republic, Slovakia and Hungary). This was reflected by the huge interest of geospatial users in CASCADOSS regional workshops, following the events in Warsaw, organized in Belgium, Poland, Czech Republic, Slovakia and Hungary in late 2008 and early 2009.

Links and references

CASCADOSS website: www.cascadoss.eu
CASCADOSS LiveDVD[1]
CASCADOSS at OSOR[2]
CASCADOSS Association[3]

M. Rusztecka
UNEP/GRID-Warsaw
www.gridw.pl

R.Wawer and T. Steenberghen
SADL, KULEUVEN R&D
http://sadl.kuleuven.be

E. Orlitova
GISAT
www.gisat.cz

A. Podolcak
Compet-Terra
www.competterra.com

[1]CASCADOSS LiveDVD: http://cascadoss.competterra.com/cascadoss.php?livedvd_en
[2]CASCADOSS at OSOR: http://bit.ly/cSYmhM
[3]CASCADOSS Association: http://cascadoss.competterra.com/cascadoss.php?home_en

Summer Training Courses on FOSS4G, 2007-2009

Jos Van Orshoven, Rafal Wawer

With support of the Flemish Interuniversity Council (VLIR) through its Short Training Initiative programme, the Spatial Applications Division of the Katholieke Universiteit Leuven in Belgium (SADL/ K.U.Leuven) has repeatedly offered two-week training sessions on the application of FOSS4G for processing and sharing of geospatial data in the context of land evaluation and land use planning. The initiatives were conducted during the summers of 2007, 2008 and 2009, and were oriented towards participants from developing countries.

Objectives

The summer courses were designed to help participants to:

- evaluate the potential of (F)OSS4G for the disciplines of physical land evaluation and land use planning;
- identify strong and weak points of FOSS4G-solutions and ways to handle them;
- acquire hands-on experience with Quantum-GIS and GRASS-software (2007–2009) and with related FOSS4G like PostgreSQL/PostGIS and R (2009 only);
- learn about the principles of OGC web mapping services (2008 and 2009);
- learn to work with MapServer and web clients: ka-map! and Chameleon (2008 and 2009); and
- learn about the technical components of Spatial Data Infrastructures (2009).

In addition, the courses were meant to provide a theoretical background regarding:

- the evolving concepts of physical land evaluation;
- the concepts and functionality of Geographic Information Systems;
- the principles of earth remote sensing for acquisition of land-related data for further processing in GIS.
- the concepts of Spatial Data Infrastructures and the European INSPIRE Directive.

The programmes were conceived in a 'Train the Trainer' spirit. An important objective was to motivate and help the participants to further spread the knowledge and skills gained.

Targeted participants, and venue

To be eligible, participants were required to have a background in the management and/or planning of natural resources (soil, water, vegetation, climate). They had to be familiar with maps and PC's, and GIS was not completely new to them. They were current or future professionals or researchers dealing with rural development and planning. They had the ambition to play a leading role within their current and/or future organisations regarding education and training in land evaluation and land use planning, GIS and earth remote sensing. The training location was the Geo-Institute of the Katholieke Universiteit Leuven (Figure 1).

Figure 1: Geo-Institute of the Katholieke Universiteit Leuven

Practical arrangements and fees

Participation was free for VLIR-UOS scholars. Scholarship also included travel and accommodation costs. 500EUR was charged to other participants from developing countries (according to the OESO/ DAC list used by VLIR-UOS) and 2420 EUR for other participants.

Campus accommodation was made available for all participants from developing countries. For scholars flights were booked and delivered by the organizers. Insurance for the duration of the course was also included for this group.

Some facts and figures

In 2007 the course was attended by 19 people from 14 countries: Belgium, Cameroon, Ecuador, Egypt, Ghana, Guatemala, Indonesia, Philippines, Rwanda, Sri Lanka, Sudan, Uganda and Zambia. Table 1 summarizes the course modules that year.

Figure 2: Participants at the workshop, 2007

The participants were chosen among 55 submitted applications. Most of the participants had strong background in research and engineering within the fields of environmental and agricultural applications, as well as in civil engineering. Through a questionnaire the participants gave a very positive feedback, proving the course materials and scheme to be very effective and suited to the participants need and expectations. However there were also a few critical opinions pointing at the lack of advanced training, especially on software programming issues.

Figure 3: Mobile participants

Figure 4: 2008 Participants

In 2008, 2 extra modules were added. The course included new material on programming and the evaluation of FOSS4G software, OSS licensing issues and OSS business models. A theoretical and practical introduction to OGC WMS and WFS and their implementation in FOSS4G software was also added. Table 2 summarizes the 2008 course. The 2008 course was attended by 25 participants from 15 countries: Bangladesh, Belgium, Bolivia, Colombia, Cuba, Egypt, Ethiopia, Ghana, Nepal, Nigeria, Philippines, Tanzania, Togo, Uganda and Zambia.

The most recent edition, conducted in 2009, put more emphasis on Spatial Data Infrastructures, introducing both theoretical and practical background, giving aspects of European regulation (INSPIRE Directive), as well as practical hands-on exercises managing databases and setting view and download services. Table 3 summarizes the 2009 offerings. The 2009 edition was attended by 21 participants, coming from 12 countries: Belgium, Bolivia, Cambodia, Ecuador, Ethiopia, Guatemala, India, Kenya, Peru,

Week	Module	Subtopics
1	Concepts, functionalities & land databases; sources of spatial data; evaluation applications of GIS;Illustrations & hands-on exercises using GRASS for Windows	Land evaluation process; need for spatial data and processing in land evaluation; GIS as information system; GIS as technology; FOSS4G vs. proprietary geomatics software; spatial data modelling & databases; sources of spatial data; query and presentation; GIS data input; geographic reference systems & coordinate transformations; topology & spatial analysis; producing maps; integration exercise; terrain modelling & surface analysis; SDIs
2	Concepts, functionalities & applications of earth remote sensing & image processing in the context of land evaluation with GIS; Illustrations & hands-on exercises using GRASS for Windows	What is earth remote sensing?; electromagnetic radiation; platforms & sensors for earth remote sensing; image visualization, preprocessing, enhancement, & transformation; classification; integration & processing of images in GIS; extended integration exercise; land cover and land use change detection; extended RS exercise; image processing & GIS

Table 1: Course programme in 2007

Week	Module	Subtopics
1	QGIS & its GRASS plugin as a FOSS4G tool for land evaluation & land use planning; Concepts & functions of Geographic Information Systems	Intro: land evaluation, geomatics, FOSS4G; GIS as information system & technology; finding, downloading & installing QGIS; QGIS & the Tabacay dataset; FOSS4G resources; modelling geographic reality with GIS; Tabacay database as model of geographic reality; viewing, query & mapping with QGIS & GRASS plugin; GIS as viewing/mapping system; coordinate reference systems; QGIS coordinate management; WMS & WFS in SDI context; QGIS as V/M system; Input of data into a gDB; GRASS database, locations & mapsets; editing & creating geodatasets with QGIS/GRASS; structures for vector geodatasets; analysis of geodatasets; analysis with QGIS GRASS plugin; input, structure, transformation & management in QGIS/GRASS; land evaluation for land use planning; terrain modelling; construction & analysis of terrain models with QGIS GRASS plugin; analytical capabilities of QGIS/GRASS; receiving/providing QGIS help/ support
	Introduction to Quantum GIS customization	Core plugins: visualization, graticules, GPS, georeferencer
		Core plugins: MapServer export; Intro to programming with Python
2	GRASS as a FOSS4G tool to integrate remotely sensed imagery & land use planning; Basics of earth remote sensing & image processing	Basics of earth remote sensing; analytical capabilities of QGIS/-GRASS; introduction to GRASS; finding, downloading & installing WinGRASS; image visualization; starting with GRASS; import/export from/slash to GRASS; image viewing & preprocessing; enhancement; classification; GRASS image processing; free image data sources; advanced integrated exercise; graduate study possibilities
	Evaluating FOSS4G software projects	Methods for evaluation; evaluating QGIS; CASCADOSS FOSS4G evaluation results; FOSS4G licensing issues; contents of KOI DVD

Table 2: Course programme, 2008

Week	Module	Subtopics
1	FOSS4G tools for processing geospatial data in the context of land evaluation and land use planning	Welcome, getting started; Intro to land evaluation, geomatics & FOSS4G; GIS: information system & technology; modelling geographic reality with GIS; finding, downloading & installing QGIS; first experience with QGIS & Tabacay geospatial database; GIS as a viewing/mapping system; coordinate reference systems; viewing, query & mapping with QGIS and GRASS plugin; QGIS coordinate management; structures for vector geodatasets; analysis of geodatasets; analysis with QGIS GRASS plugin; terrain modelling; construction & analysis of terrain models with QGIS GRASS plugin; finding, downloading & installing R statistics software; integrated exercises with QGIS, GRASS plugin & R; input of data into a gDB; editing & creating geodatasets with QGIS/GRASS; QGIS-plugins; receiving/providing help/support regarding QGIS; QGIS as V/M system; input, structure, transformation & management capabilities of QGIS; analytical capabilities of QGIS; earth remote sensing; introduction to GRASS; finding, downloading & installing GRASS; first experiences with GRASS
2	FOSS4G tools for sharing geospatial data	Sharing geospatial data with SDIs; SDI services for sharing; concepts & exercises of metadata services, webmapping services, and web feature services; object-relational database management systems (ORDBMS) to improve accessibility of geospatial data in multi-user environments; PostgreSQL / PostGIS as a FOSS ORDBMS; grad study opportunities; documentation & evaluation of FOSS4G solutions; business models for FOSS; FOSS licenses

Table 3: Course programme, 2009

Tanzania, Uganda and Zambia. In 2009 VLIR-UOS changed the list of developing and low-income countries, eligible for scholarships, which was reflected in the decrease of the countries represented in our course.

Final remarks

We have provided participants of our courses with several questionnaires, starting from so called "zero measurement", where we assessed the knowledge and skill in ICT, GIS, RS, SDI of our participants at the beginning of the course. Zero measurement was followed by evaluation questionnaires for the modules as well as evaluation of the whole course, where we emphasized the future plans for further use of the software and dissemination of the materials. Then we sent additional questionnaires, some 6 months after the course, asking about the actual use of the materials and software in the daily professional tasks of the participants. The results from 2007 and 2008 assessments, compared to our previous courses, where commercial and closed software was used, were presented during AGILE conference in 2009 and are published in the proceedings:

Van Orshoven, J., R. Wawer and K. Duytschaever, 2009. Effectiveness of a train-the-trainer initiative dealing with free and open source software for geomatics. CD-ROM-proceedings (J.-H. Haunert, B. Kieler, and J. Milde, eds. (2009)) of the 12th AGILE International Conference on Geographic Information Science (2009), Hannover, Germany, 2–5 June 2009, IKG, Leibnitz Universität, ISBN 2073-8013.

The paper is also available on the AGILE 2009 website: http://www.ikg.uni-hannover.de/agile/fileadmin/agile/paper/136.pdf

Jos Van Orshoven
Rafal Wawer

Topical Interest

Why Every Open Source Project Needs a Good Dictator

How A Good Dictator Can Solve Problems With The Open Source Software Development Model

by Landon Blake

Introduction

This article examines the need for a 'good dictator' in open source software projects. It begins by describing the problems that can result in an open source project that lacks a good dictator, proceeds by explaining how a good dictator can help solve these problems, examines the qualities of a good dictator, and concludes with a description of the type of open source projects that would benefit the most from a good dictator.

Problems Resulting From a Lack of Clear Leadership

There are several problems that can plague an open source leadership that lacks clear leadership. These problems include:

Stagnation in development of the open source software that is developed by the project. Symptoms of this problem include infrequent releases and a lack of new features. Stagnation can result when there are no clear goals for the program or an overall push to take the program in a forward direction.

A 'muddied' code base that suffers from source code contributions with an inconsistent coding style. This coding style could be the formatting and structure of the actual source code, or the way the architecture of the program is designed and implemented.

Inconsistent branding of the open source project. This includes the design of logos, designation of text fonts and color schemes. Inconsistencies in these elements show up in both the documentation and the graphical user interface of the software produced by the project.

What is a Dictator?

Before we can describe how a dictator can solve the problems listed previously, it is important that we define the term dictator for the purposes of this article. A dictator is a person or organization that provides strong leadership of an open source software project. This leadership often comes in the form of an individual programmer, team of programmers, or organization that has specific goals for the open source project and a willingness to make decisions that may displease certain segments of the community.

The Best Type of Dictator

What makes a good dictator for an open source software development project? A good dictator has the following characteristics.

A good dictator has a long-term commitment to the project. (For example: A company has a long term commitment to a software project because it distributes the software for a hardware platform that it produces. Another example: A non-profit organization that uses an open source program as a key part of its operations.)

A good dictator knows how to balance competing interests. This includes knowing what is worth fighting over, knowing how to best handle difficult people, and being able to tell contributors 'no' when it becomes necessary.

A good dictator has a large investment in the open source software project, but maintains an ability to see beyond their own immediate needs.

Solutions Provided By a Good Dictatorship

How does an open source software project benefit from a good dictator? How does a good dictator fix some of the problems that arise when an open source software project lacks good leadership?

A good dictator can set programming, marketing, and documentation priorities for a project. A good dictator can outline a development road map and set a release schedule.

A good dictator can set and enforce coding styles. This makes the structure and formatting of the actual source code more consistent. It also makes decisions about the implementation of the program architecture more consistent.

A good dictator can set and enforce standards for branding of the product. This can result in a clear image of the program's 'brand' and a more effective marketing effort for the open source project.

Landon Blake
Redefined Horizons
www.redefinedhorizons.com
sunburned.surveyor AT redefinedhorizons.net

Programming Tutorials

GPGPU With GDAL

Basics of GPGPU Interfacing

By Yann Chemin

Introduction

The new generation of graphic cards have Graphical Processing Units (GPU) installed on-board. These GPUs commonly have hundreds of processing cores, high speed parallel architecture, and RAM in the Gigabyte range. Development of GPUs was driven primarily by the processing power required to create and display virtual reality environments in the gaming industry. Now they are used to compute general physics accelerated algorithms for environmental modelling like fluids dynamics.

General-Purpose computation on GPUs (GPGPU) is a relatively new type of computation made possible by the increasingly varied types of computations available on those graphic cards. They are called "coprocessors" in computer engineering. Their high-speed high-parallel architecture makes them very attractive for computations requiring large amounts of operations on each dataset units.

The main point of GPGPU is to manage the memory allocation in the GPU itself. To do this, copy the data into the GPU memory before processing and copying it back to the computer resources outside of the GPU after processing is complete.

In this article we will consider an example of GPU programming. In this example we will use a language for NVIDIA GPUs called Compute Unified Driver Architecture (CUDA).

GPGPU Programming Example

This is a C/C++ language addendum that enables the code to send your data to your GPGPU-enabled NVIDIA graphic card for processing, after which you can retrieve your results from the graphic card RAM and send it to the computer hard disk.

For your computer to understand this code, it needs an additional compiler to be used with your C/C++ compiler. This is not your standard C or C++ compiler. The NVIDIA compiler can be downloaded from http://www.nvidia.com, Look for it in the "CUDA Zone". Read the documentation for your graphic card's driver to see if it already has the CUDA functionality. ION-based laptops are CUDA enabled too.

A CUDA-enabled file `cuda_ndvi.cu`, can be compiled with the NVIDIA compiler (nvcc) like this:

```
nvcc -o ndvicu cuda_ndvi.cu
```

A CUDA-enabled file can be run like this:

```
./ndvicu
```

If you don't have a CUDA capable GPU, compile it in emulation mode:

```
nvcc -deviceemu -o ndvi_cu ndvi_cuda.cu
```

Example Makefile:

```
ndvicu: cuda_ndvi.cu
        nvcc -o ndvicu cuda_ndvi.cu \
        -I/usr/include/gdal/ -L/usr/lib \
        -lgdal1.6.0
```

A typical raster program first loads datasets, define raster data holders for our Red (red), Near Infrared (nir) and Normalized Difference Vegetation Index (ndvi) image bands with GDAL. Then it loads the red and nir bands into line buffers. Using line buffers here is the choice, since the processing is not col/row dependent. GPUs have clear and easy ways to access and work with 2D and 3D pixel localization access within the (GPU) RAM matrix.

First we allocate arrays on the host (our computer itself) so we can fetch the data from our images.

```
//N = col x row
int N=nXSize*nYSize;
float *red=(float *)malloc(sizeof(float)*N);
float *nir=(float *)malloc(sizeof(float)*N);
float *ndvi=(float *)malloc(sizeof(float)*N);

//Load input datasets
GDALRasterIO(hBandRed,GF_Read,0,0,\
   nX,nY,red,nX,nY,GDT_Float32,0,0);
GDALRasterIO(hBandNir,GF_Read,0,0,\
   nX,nY,nir,nX,nY,GDT_Float32,0,0);
```

On the GPU side, we start by allocating variables with the suffix `_d` (d for device) to remind us of their location of use, in the GPU device. As everything is a grid (2D or 3D matrix) in a GPU, we allocate an integer N as our image total dimension, this is actually the total length of the grid allocated inside the GPU (it could be written like this: `N=row_d x col_d`).

```
/* pointers to GPU device memory */
float *red_d, *nir_d, *ndvi_d;

/* Allocate arrays on GPU device*/
cudaMalloc((void **)&red_d,sizeof(float)*N);
cudaMalloc((void **)&nir_d,sizeof(float)*N);
cudaMalloc((void **)&ndvi_d,sizeof(float)*N);
```

Once the data is in our computer memory and GPU memory has been prepared to receive the data, it is time to send the row data into the GPU. To do that, we have to use a specific function called:

```
cudaMemcpy(GPU_memory,PC_memory,\
  size_of_data,cudaMemcpyHostToDevice)
```

The last argument indicates the direction of the copy of the data. In this case we send the data from the computer to the GPU, so the direction is `cudaMemcpyHostToDevice`, to retrieve the data after computation, we will use the opposite direction `cudaMemcpyDeviceToHost`.

```
/* Copy data from CPU host to
   GPU device memory */
cudaMemcpy(red_d,red,sizeof(float)*N,\
    cudaMemcpyHostToDevice);
cudaMemcpy(nir_d,nir,sizeof(float)*N,\
    cudaMemcpyHostToDevice);
```

When the data has reached the GPU memory, it is time to apply calculations on it. This is done in GPGPU computing by applying a kernel. The incantation for doing so is shown below:

```
/* Add arrays red, nir and
   store result in ndvi */
ndvi_cu<<<dimGrid,dimBlock>>>(red_d, \
     nir_d, ndvi_d, N);
```

In this case, the kernel to be used is called `ndvi_cu`, it is applied on a grid. That grid is sub-divided in blocks, which are the units of work allocation within the GPU. The grid in our example is N, our row-x-col image dimension. It is visualized as a grid by the GPU and is split into computing blocks limited by the architecture of the GPGPU.

At the time of the writing of this document, lower-end graphic cards are capable of 256 or 512 size blocks. The bigger the block, of course, the less number of job distributions to complete the processing of our row data. If the row data has a size of N=nXSize*nYSize=1024, it will take 4 blocks of 256 or 2 blocks of 512 to compute it. As you can see, Block and Grid allocation are defined into dim3. This is because GPUs, being graphical devices, operate natively in 3 dimensions.

```
/* Compute blocks of data to send to GPU */
// On GeForce 8600 Galaxy x=256
// On GeForce 9500 Galaxy & 9800 GT x=512
int x=512;
dim3 dimBlock(x);
dim3 dimGrid((N/dimBlock.x)+ \
     (!(N%dimBlock.x)?0:1));
```

The kernel was initially a standard C function. It was modified to appropriately take benefit of the GPGPU architecture.

```
__global__ void ndvi_cu(float *red, \
   float *nir,float *ndvi, int N)
{
 int i = blockIdx.x * blockDim.x \
       + threadIdx.x ;
 if ( i < N )
  ndvi[i]=(nir[i]-red[i])/(red[i]+nir[i]);
}
```

Synchronize the data movement with the job completion in the GPU to streamline the processing. This is done by applying the function `cudaThreadSynchronize()`:

```
/* Block until device completed processing */
cudaThreadSynchronize();
```

From this point forward, data is copied back to the computer from the device using the `cudaMemcpy` function. You can then free the memory on the GPGPU device.

```
/* Copy data from GPU device
   to CPU host memory */
cudaMemcpy(ndvi,ndvi_d,sizeof(float)*N, \
           cudaMemcpyDeviceToHost);
cudaFree(red_d);
cudaFree(nir_d);
cudaFree(ndvi_d);
```

After this point, our program will look like a standard C program, as seen earlier, where GDAL takes over the row data and writes it to the disk. We then close the output file and free the memory.

```
GDALRasterIO(hBandNdvi,GF_Write,0,0, \
   nX,nY,nir,nX,nY,GDT_Float32,0,0);
if( red != NULL ) free( red );
if( nir != NULL ) free( nir );
if( ndvi != NULL ) free( ndvi );
GDALClose(hDatasetRed);
GDALClose(hDatasetNir);
GDALClose(hDatasetNdvi);
```

Conclusion

This article is a short path to implementing GDAL-based raster processing in the scalable and highly parallel environment of GPUs.

More information on GPGPU in general can be found at http://www.gpgpu.org.

Yann Chemin
International Centre of Water for Food Security
http://www.csu.edu.au/research/icwater
ychemin AT csu.edu.au

Case Studies

gvSIG is a viable robust alternative to commercially available GIS packages

by Simon Cropper

Abstract

I have been compelled to comment on the latest release of gvSIG (Version 1.9, Build 1253) due to the fact that it is the first Open Source Geographical Information System that I have encountered that has allowed me to fulfill the full complement of workflow processes for a typical project without having to use third party software or falling back to commercial software.

Context

To provide some context to this statement I wish to provide some background on myself and my needs. I am an environmental consultant conducting flora and fauna surveys within Southeastern Australia. Clients range from local, state or national government agencies or land developers addressing their legislative obligations. Every job requires the acquisition and creation of geospatial data, simple geospatial analysis and the preparation of maps. Every job has variations on the theme with clients providing information in various datums and formats. Output is relatively constant with maps being used as JPEGs in reports or supplied as shapefiles to the client. The typical workflow for a job is outlined in Table 1.

Background

Over the last 15 years I have used ArcView with a myriad of third party extensions and scripts, but have over the last year been searching for a reliable and robust alternative. Essentially I needed a package that could implement the workflow outlined in Table 1, with little or no need to export and manipulate in other packages and if that was necessary there was no need to export the data into other formats (i.e. you can manipulate the same data files rather than have multiple versions of the same information lying around).

Over the last year I have tried various versions of

- QGIS
- Grass
- Kosmo
- OpenJUMP

and others to carry out what I needed. I also tried early versions of gvSIG but found some basic requirements were not met. Beta versions of 1.9 looked promising but they were unstable with regular crashes making it difficult to justify the time to acquaint myself with the program.

As you can see from Table 1 the primary failings of the alternative packages was the ability to natively view ECW files, being able to reproject data into new Spatial Reference Systems and map production. All were variously capable of viewing and manipulating

	gvSIG+	QGIS	Grass	Kosmo	OJUMP+
STAGE 1 – DATA ACQUISITION					
Acquire aerial photography of study area and vector data showing existing features. Only gvSIG and Kosmo had native support for ECW files.	Y	Pd	Pd	Y	Yd
Manipulate the various data sets so they all overlap in the appropriate Spatial Reference System (i.e. reproject vector layers).	Y	N	Yd	N	N
STAGE 2 – PREPARATION FOR FIELDWORK					
Identify extent of study area based on cadastral data and the objectives of the client.	Y	Y	Yd	Y	Y
Extract data from state government managed databases and geospatial libraries of environmental data like flora, fauna, vegetation, geology, wetlands, etc. Incorporate into project. Includes the need to import tables as Event Layers.	Y	Y	Yd	P	P
Stratify study area based on aerial photography interpretation, contours, soils and vegetation data (data just represented visually, stratification done manually).	Y	Y	Yd	Y	Y
STAGE 3 – FIELD WORK					
Adjust boundaries of strata based on field observations.	Y	Y	Yd	Y	Y
Mark extent of any significant plant population or animal habitat on base map using features visible on aerial photography.	Y	Y	Yd	Y	Y
STAGE 4 – ANALYSIS					
Clean up the vector data created in the field.	Yd	N	Yd	Y	Y
Calculate and store area of each stratum in attribute table.	Y	N	Yd	N	N
Collate landscape, neighborhood and other miscellaneous attributes for each stratum (i.e. direct data entry into tables and joins).	Y	N	Yd	N	Y
STAGE 5 – MAP PRODUCTION					
Create basic maps showing the results of the analysis – the map should have grid lines showing the SRS of the View.	Y	P	?	P	N
Export map into format that can be directly imported into a Word Processor.	N	Y	?	N	N

Table 1: An outline of the workflow for a typical flora and fauna survey by Botanicus Australia Pty Ltd and whether certain FOSS GIS Desktop Packages were capable of completing the tasks. Y = capable. N = not capable. P = partly capable. d = difficult to achieve (either not intuitive or needs workaround). gvSIG+ = gvSIG 1.9 + Sextante 0.3. QGIS=QGIS 1.3.0 Mimas . GRASS = WinGrass 6.3. Kosmo=Kosmo 1.2.1. OJUMP=OpenJUMP 1.3 with Sextante 0.3.

vector data stored in common vector formats. Packages varied considerably in how intuitive the interface was and their stability.

Conclusion

At the time that this article was being prepared I have been using gvSIG 1.9 (Build 1253) in production for several months. The following points outline my initial observations on its use.

1. The transition from ArcView to gvSIG was seamless with most functions being found in similar locations. I had a similar experience when trialling Kosmo, but not with any of the other packages mentioned. OpenJUMP is intuitive but limited in functionality.
2. gvSIG uses ECW, DWG, DXF, shapefiles – all file formats I regularly encounter.
3. gvSIG can convert from AGD66 GEO/AMG55 to GDA94 GEO/MGA55 – something I need to do regularly.
4. gvSIG could handle quite a lot of data,files, annotations, etc. The only time I noticed the system slowing down was in producing a map. In this window, gvSIG was slow but it did not crash.
5. gvSIG crashed a few times. These instances appear to be related to small or peculiar bugs not captured by the program. These issues are being addressed by the developers relatively quickly. In comparison with ArcView this is quite good. For some largish projects ArcView would crash 3–4 times a day. It is worth noting here that I was using ArcView 3.1, which is not currently supported and is struggling to keep up with operating system changes (XP service packs, patches, etc).
6. I am keen to eventually migrate from Windows XP to Ubuntu or Debian, so am keen to ensure the system I use is suitable for these operating systems. gvSIG does, so this is great.
7. I still think the map production facility in gvSIG can be improved – especially the output. My needs are primarily generation of a file than can be imported into OpenOffice Writer. I found I got a much better output file for small maps in landscape by capturing the screen and manipulating in GIMP, than using any of the standard output options provided by gvSIG. With larger maps or maps in portrait I have found ImagePrinter to capture printer output works really well. Something worth noting here was the discovery that the graticules in gvSIG are actually dynamic. Once set up you can pan the underlying map and the graticule changes - its like looking through a window. This is a very useful and long awaited functionality I was looking for in a replacement to ArcView.
8. Coupled with Sextante, gvSIG captures most of the tasks that I have done over the last 15 years.
9. I have not been able to find any good tutorials on how to use the command box when editing a shapefile or the JPython Console. These features look very promising but without some sort of tutorial, examples or manuals it is impossible to evaluate these tools. In my mind this is the only area where ArcView 3.1 is still better than any open source alternatives as it has quite a large and easily navigated script library which allows for people to contribute scripts and extensions for manipulation of spatial data that can be downloaded by anyone.

The overall result of this trial is that I found the gvSIG loaded with the Sextante extension was able to complete all the steps in my work flow diagram except *Export map into format that can be imported into Word Processor* but the use of GIMP or ImagePrinter is a quick workaround. This is better than any other open source package I have trialled over the last year. Coupled with the myriad of additional tools bundled in this package and additional extensions that can be downloaded this version of gvSIG has left me smiling. I can do nearly all things that I want and have lots more to explore and evaluate for use in my business activities. I would like to congratulate the developers and the Generalitat Valenciana on a fantastic product. I look forward to being part of this dynamic open source community, and helping further the development of this product.

Simon Cropper
Botanicus Australia Pty Ltd
PO Box 160
Sunshine, Victoria, Australia 3020
scropper@botanicusaustralia.com.au

Project Introductions

GRASS Image Processing Environment

Application to Evapotranspiration Direct Readout

Yann Chemin, Thomas Alexandridis and Ines Cherif

Abstract

Satellite imagery provides a large amount of useful information. To extract this information takes understanding, processing and time. In GRASS GIS, a given professional of image processing can develop his/her own modules for processing satellite imagery. The aim of this paper is to communicate the main lines of the development of an automated Landsat 7ETM+ chain processing using modules of the GRASS Image Processing Environmment (GIPE). Specifically called Direct Readout [1], the target is real-time processing of Level 1B standard imagery (Digital Number) into evapotranspiration products (considered level 4 in MODIS).

Introduction

Satellite imagery is spatial information, it has spatial dimensions, but also radiometric, spectral and temporal resolutions. There is a large amount of data in this raw material already, but this needs quantification for supporting decision. Processing of satellite imagery is the science of bringing measurements, units to the satellite image pixel. These measurements are commonly found in terms of indices (vegetation indices especially), some more physically-based information can be derived and units are then starting to appear, reflected radiations in Watt per square meter (W/m2) are the first to be generated from the raw data received. Some more complex calculations, some would say modelling, can actually bring more tangible information for decision-makers. In this paper, we will deal with water evapotranspired, in terms of mm/day. This kind of information is critical in watershed analysis, water balance, and crop production[2][3].

Evapotranspiration monitoring is on a high priority on several decision-making entities agendas. Among those entities are various water resources management-related agencies, publicly or privately run. Of course, irrigated crop and environmental managers have dire interest in accurate, spatially discreet, standardized and temporally available water balance inputs. Also, natural resources managers use evapotranspiration for assessing agricultural water pressures and desertification risk[4][5]. For about 30 years, the steady rise of methods/models in satellite remote sensing has increasingly permitted easier and/or more precise calculations of ET from satellite imagery. As of the 1990s, thermo-dynamically-based energy balance models have been given some encouraging results, some reviews can be found in [6], [7], [8] and [9].

This paper is presenting the application of a *suite* of various GRASS GIS modules for the image processing requirements of evapotranspiration mapping. The *suite* was dubbed GRASS Image Processing Environment (GIPE), as it was created with the aim of fulfilling any step of the processing requirements independently of each other, if it is so desired by any future user for any other future processing needs. This explains the number and diversity of the modules developed within this GIPE.

Objectives

The main objectives are:
- Overview the GIPE
- Introduce the Landsat 7 ET script generator
- Produce some output of ET

Overview of the GIPE

The need of creating a set of modules in GRASS GIS to perform the regular satellite image processing tasks is required for any production or research work. The Grass Image Processing Environment tries to provide such tools at the atomistic and factory levels. Respectively, there should be a module for each step of any of the satellite imagery processing, but for some repetitive tasks when scripting can become too heavy, one module should provide the direct high-level output expected without losing too much on parameterization.

The functions developed were essentially in a few categories, pre-processing and products calculations. Respectively, preparation of raw satellite imagery into usable geodata (mostly radiometric correction) and preparation of given products like NDVI, Albedo, and higher-level ones like ET or soil moisture. In Figure. 1, a list of various developed modules is shortly described.

Module	Description
i.albedo	Albedo (0.3-3micro.m)
i.biomass	Biomass growth (kg/day)
i.dn2full.l5/7	Landsat5/7 DN to Reflectance/Temperature
i.dn2potrad.l5/7	ET: Landsat5/7 DN to ET potential
i.dn2ref.ast	Aster DN to Reflectance
i.dn2ref.l7	Landsat7 DN to Reflectance (visible only)
i.emissivity	Generic emissivity for the thermal bands
i.evapo.MH	ET: Reference ET Hargreaves (2 types)
i.evapo.potrad	ET: Potential ET (radiative method)
i.evapo.PT	ET: Potential ET Priestley-Taylor
i.evapo.SENAY	ET: Actual ET Senay (2007)
i.latitude	Latitude map generation
i.longitude	Longitude map generation
i.sattime	Satellite overpass time map generation
i.sunhours	Sunshine hours (potential)
i.vi	Vegetation Indices (13 types)

Figure 1: GIPE modules listing: Various

Modules	Description
i.eb.deltat	Surface-air temperature difference
i.eb.disp	Displacement height
i.eb.eta	Actual ET
i.eb.evapfr	Evaporative Fraction
i.eb.g0	Soil Heat Flux
i.eb.h0	Sensible Heat Flux
i.eb.h_iter	Sensible Heat Flux Pawan04 model
i.eb.h_SEBAL95	Sensible Heat Flux SEBAL95 model
i.eb.molength	Monin-Obukov Length
i.eb.netrad	Net Radiation
i.eb.psi	Psichrometric factors
i.eb.rah	Aerodynamic resist. to heat flux
i.eb.rohair	Atmos. Air Density
i.eb.ublend	Wind speed@blending height
i.eb.ustar	Nominal Wind Speed
i.eb.z0m	Roughness length

Figure 2: GIPE modules listing: Energy Balance

Mostly Figure. 1 refers to known processing of satellite imagery. the module *i.vi* is also (but not in this table) available in parallel (*i.vi.mpi*) and in grid (*i.vi.grid*) developed for educational purpose in [10]. Additionally, it can be noticed in Figure. 1, there are many ET map generation modules. Three main groups of ET maps are available here. The first one is called the *Reference ET*, calibrated after a 20 cm reference height grass without water stress. The second one is the *Potential ET*, using all energy available in evapotranspiration, considering unlimited availability of water. At this point it is important to note that the GRASS HydroFOSS Add-on [11] is also having an ET module following the *Potential ET* methodology from Penman and Monteith (*h.evapo.PM*) that GIPE also installs by default. Finally, *Actual ET* is a more complex assessment of the environmental water stress to provide information on water that *actually* vaporized from water bodies and soil or transpired from vegetation.

Figure. 2 provides with specific energy balance modules created for the purpose of calculating actual evapotranspiration using mainly equations from [12] and [13]. Those are in fact two versions of the Surface Energy Balance Algorithm for Land (SEBAL) as named by [13].

Processing ET

Processing a satellite image for evapotranspiration is overviewed in Fig. 3

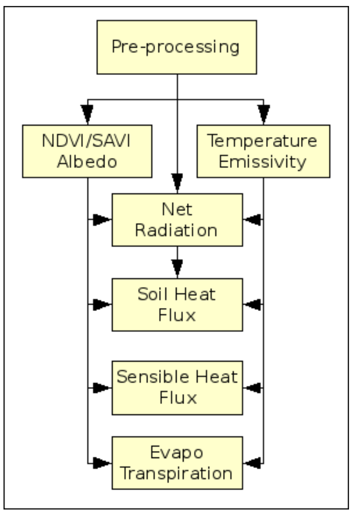


Figure 3: ET Processing Overview

Script generator: Landsat 7ETM+

The Landsat 7ETM+ script generator is a small program included in GIPE that creates a full processing of various ET maps by auto-configuring all the steps from importing/calibrating landsat bands up to the final products. Below is a small sample of the script after generation. It loads the .met metadata file, reads parameters from it, and then fills up the GIPE modules pertaining to the pre-processing and higher level processing of the data.

Ungzip all Landsat bands, import/create new location, set region to temperature map. *DN to Radiance to Reflectance Top of Atmosphere:*

```
i.dn2full.l7 metfile=pXXrXX.met
output=pr
```

[skipping...Albedo, Pre-ET Potential, Download-/Import SRTM, Slope/Aspect...]
Potential ET:

```
i.evapo.potrad -r albedo=pr.albedo tempk=pr.61 \
 lat=pr.latitude doy=pr.doy tsw=pr.tsw etpot=pr.etpot
rnetd=pr.rnetd
```

NDVI, SAVI:

```
i.vi viname=ndvi red=pr.3 nir=pr.4
vi=pr.ndvi
```

[skipping...Pre-ET stuff, Net radiation,...]
Soil heat flux:

```
i.eb.g0 albedo=pr.albedo ndvi=pr.ndvi tempk=pr.61 \
 rnet=pr.rnet time=pr.sath g0=pr.g0
```

[skipping...Displacement height, Atmospheric air density,..]
Sensible heat flux [12]:

```
   i.eb.h_iter rohair=pr.rohair cp=1004.16 dtair=pr.delta
tempk=pr.61 disp=pr.disp z0m=pr.z0m z0h=pr.z0h  u2m=u2
h0=pr.h0
```

Soil Moisture:

```
i.eb.evapfr -m rnet=pr.rnet g0=pr.g0 h0=pr.h0 \
 evapfr=pr.evapfr theta=pr.theta
```

Actual ET:

```
i.eb.eta rnetday=pr.rnetd evapfr=pr.evapfr \
 tempk=pr.61 eta=pr.eta
```

Experimental Results

The study area is in the area of Strimonas river basin in Greece [14]. Strimonas river basin is a transboundary river basin situated in north Greece (Serres). A large reservoir (Lake Kerkini) had been constructed in the 1930's to store and regulate water in the Greek part, and has now become a wetland of International Importance by the Ramsar Convention, and is included in the Natura 2000 network. An extensive irrigation system taps water from the reservoir and river diversion dams, and distributes water using open canals. Local groundwater pumping stations provide supplementary water to account for the inefficient operation of the irrigation system. Irrigated area exceeds 100,000 ha, dominated by rice, maize and cotton.

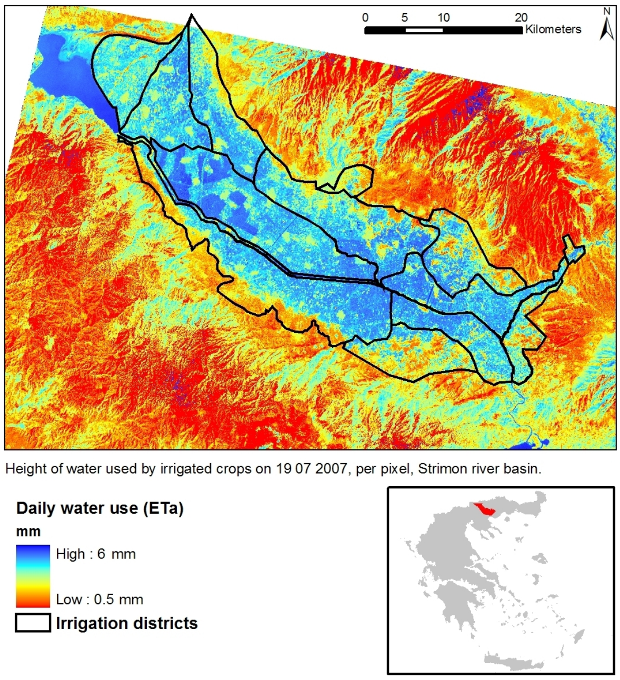


Figure 4: ETa map of Strimonas Basin after SEBAL95

Actual ET

A first SEBAL processing by [13] yielded to an evapotranspiration map (v1995, using i.eb.h_SEBAL95 to estimate sensible heat flux) in Fig.4, similarly, another map was created following SEBAL version as seen in [12] (using i.eb.h_iter instead). The determination of the sensible heat flux by iteration being the specificity of this model, the main module used has been cited above. Another module (i.eb.h0) is an atomic equation that does not perform anything else than h0 calculation. Using shell scripting and i.eb.h0 and other i.eb.*, one can reconstruct complex models like i.eb.h_SEBAL95 or i.eb.h_iter or any other at their will.

A different location, southwest Southwestern of Strimonas study area, is also used to demonstrate the comparison capacity of having several models targeting the same processing into a single processing GRASS environment. A disparity map (in Fig.5) was made as [ETa_v1995 - ETa_v2004] in mm/day of difference across the two versions. One can see that if the map is positive, then the ETa_v1995 version is overestimating and if negative, ETa_v1995 is underestimating.

Additionally, a histogram (in Fig.6) of the disparity map shows the amount and value of the disparity in terms of pixel count and ETa difference (mm/day). The equation is disparity_eta=[eta_SEBAL95 - eta_Pawan04], which incidentally has a lot to do with the way both estimate h0 parameter through (i.eb.h_SEBAL95 or i.eb.h_iter).

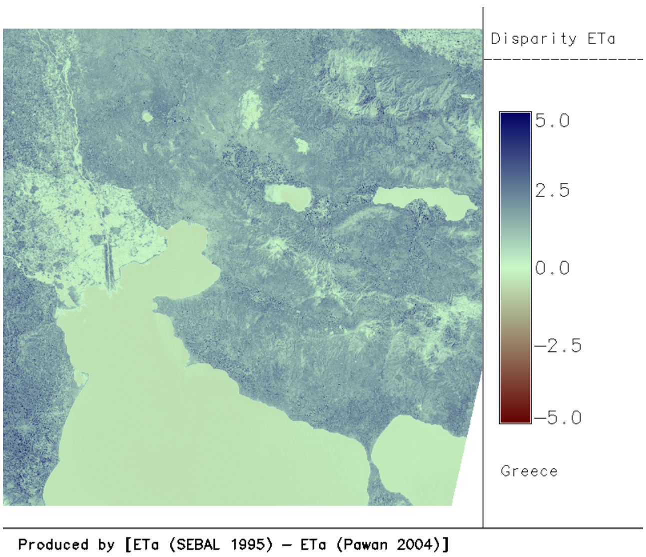


Figure 5: ETa disparity map

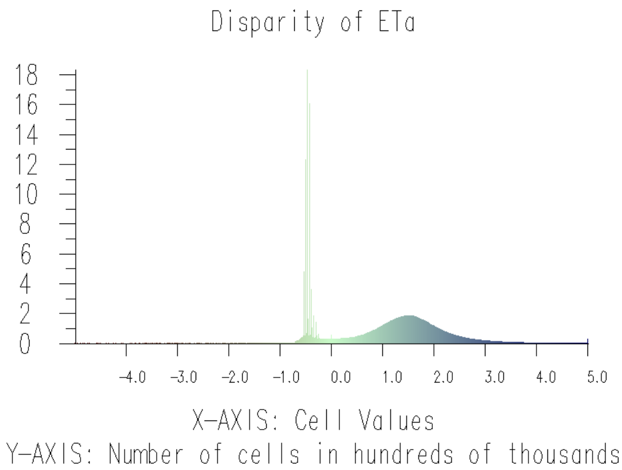


Figure 6: ET disparity histogram (v1995-v2004)

One can witness in Fig.6 that both methods being from the same origin, yet the newer implementation by [12] (esp. i.eb.h_iter) differs from the original [13] (see i.eb.h_SEBAL95). The high peak on the negative side near 0.0mm/d difference is the water pixels estimations, which are quite concurrent. However the smooth distribution on the positive side indicates that SEBAL from Bastiaanssen in 1995 has higher land-based estimations than the version of Pawan in 2004. This is certainly due to the re-assessment of the [TsoilTair] and aerodynamic roughness of momentum heat (rah) during the iterations of i.eb.h_SEBAL95, which do not occur in i.eb.h_iter [12].

Conclusion

The GRASS Image Processing Environment (GIPE) is a set of Add-Ons for GRASS GIS to process various satellite imagery into higher-level products. Landsat example above is but one of regular satellite imagery that can be (pre)processed in GIPE. Currently, specific satellite pre-processing found in GIPE includes the following sensors: ASTER, MODIS, AVHRR, Landsat 5/7. Higher-level processing is sensor independent, as long as spectral or temporal requirements are met. Finally, there are additional modules not listed here, mostly because of unfinished state to the day of this reporting.

Acknowledgements

This work was funded by the extension of the European Space Agency GSE-Land project (http://www.gmes-gseland.info/) in Greece. Infoterra GmbH is gratefully acknowledged for their support.

Yann Chemin - ychemin AT csu.edu.au
International Centre of Water for Food Security
http://www.csu.edu.au/research/icwater

Thomas Alexandridis - thalex AT auth.gr
Ines Cherif - icherif AT yahoo.com
Agricultural University of Thessaloniki
http://users.auth.gr/thalex

Bibliography

[1] http://www.conabio.gob.mx/conocimiento/premota/doctos/home.html

[2] Chemin Y., Platonov A., Ul-Hassan M. and Abdullaev I. (2004) Water depletion assessment at administrative and irrigation levels. Case Study of Ferghana Province using public remote sensing data. Agricultural Water Management 64(3):183-196.

[3] Chemin Y., Platonov A., Abdullaev I. and Ul-Hassan M. (2005) Supplementing farm level water productivity assessment by remote sensing in transition economies. Water International 30(4):513-521.

[4] Alexandridis, T.K., Takavakoglou, V., Crisman, T.L. and Zalidis, G.C. (2007) Remote sensing and GIS techniques for selecting a sustainable scenario for Lake Koronia, Greece. Environmental Management 39:278-290.

[5] Silleos, N., Cherif, I., Kolejka, J., Karydas, C., Katsogiannos, F. and Silleos, G. (2008) Desertification Assessment in the Lesvos Island in Greece and Methodology for Mitigation Action Planning. Proceedings of the International Advanced Workshop on Information and Communication Technologies for Sustainable Agri-production and Environment (AWICTSAE 2008), Alexandroupoli, Greece, 22-23 May, 2008.

[6] Kustas, W.P., and Norman, J.M. (1996) Use of remote sensing for evapotranspiration monitoring over land surfaces. Hydrological Sciences - Journal - des Sciences Hydrologiques 41(1):495-516

[7] Bastiaanssen, W.G.M. (1998) Remote sensing in water resources management: the state of the art. International Water Management Institute (IWMI), Colombo, Sri Lanka

[8] Menenti, M., (2000) Evaporation. In Remote Sensing in Hydrology and Water Management, Gert A. Schultz and Edwin T. Engman (Eds.) Springer Verlag, Berlin, Heidelberg, New York 157-188.

[9] Courault D., Seguin B., Olioso A. (2005) Review on estimation of evapotranspiration from remote sensing data: From empirical to numerical modeling approaches. Irrigation and Drainage Systems 19:223-249.

[10] Akhter, S., Chemin, Y., Aida, K. (2006) Satellite Image Processing on Distributed Computing Environments HPCS Conference, Japan, 20 December 2006.

[11] Cannata, M. (2007) HydroFOSS: A GIS embedded approach for Free & Open Source Hydrological modeling http://istgis.ist.supsi.ch:8001/geomatica/index.php?id=1

[12] Pawan, K.S. (2004) Diagnosing irrigation water resources with multi-sensor remote sensing and GIS techniques : a case study of the Roxo dam irrigation system, Portugal M.Sc. Thesis, ITC, The Netherlands

[13] Bastiaanssen, W.G.M. (1995) . Regionalization of surface flux densities and moisture indicators in composite terrain. A remote sensing approach under clear skies in Mediterranean climates. Agricultural Research Department, Wageningen, The Netherlands Report 109.

[14] Alexandridis, T., Chemin, Y., Cherif, I., Tsakoumis, G., Galanis, G., Zalidis, G., Silleos, N. (2008) Improving spatial resolution of agricultural water use estimations using ALOS AVNIR-2 imagery. ALOS P.I. Symposium, Rhodes, Greece, 3-7 November 2008

SEXTANTE, a Free Platform for Geospatial Analysis

by Olaya, Victor

Introduction

The SEXTANTE project aims to create a platform for the development of geoalgorithms that makes it easy both to implement and to use those algorithms. By doing that, SEXTANTE wants to become a reference element for geospatial analysis in the Java GIS world. Currently, more than 230 extensions have been developed so far. SEXTANTE is being developed by the Junta de Extremadura (local government of Extremadura, Spain) to fulfil their needs in terms of geographical analysis. This follows the current line of work in the region, which started with the largely acclaimed Linex (a custom Linux distribution) and supports free software through the development of new free tools.

A bit of history

The SEXTANTE project was launched in 2004 with the main goal of developing a GIS solution specially designed for the needs of regional government foresters. Though it was originally targeted at professional of forest management, it has proved to be an all-purpose solution suitable for any user in need of strong geospatial analysis capabilities, and it is developed as such nowadays. Additional elements for forest inventory are being developed as well within the project, but they will not be covered in this article. The first version of SEXTANTE was based on the German software SAGA2 , a GIS mainly focused on analysis. The original SAGA set of 120+ analysis modules was enriched by more than 70 new ones, and some modifications were also made to the core of the system. A very close relation existed between the SAGA and the SEXTANTE teams, and both the extensions and the core modification eventually made their way to the SAGA official distribution and are nowadays included in current SAGA releases. By that time, gvSIG was not yet a mature GIS product, and it was deemed unsuitable for the goals of the project. However, gvSIG soon experienced an impressive growth and quickly became a full-fledged GIS, including many features not found in SAGA, such as connections to Web services. The decision was taken to migrate all the previous work and apply all the expertise acquired in our work with SAGA to turn gvSIG into a powerful geospatial analysis tool. Although rich in operations, gvSIG had a lack of analysis functions (except for a small set of geoprocesses for vector layers, including operations like buffering, cut, merge and join, among others), so the result would be beneficial for both parties. The following steps were taken to develop the gvSIG version of SEXTANTE:

1. Creating a base layer on top of which extensions for geospatial analysis could be easily implemented. That would encapsulate the complexity of the of the gvSIG extension and plug-in architecture, and make it easier to implement new geo-algorithms, following the ideas of SAGA.
2. Migrating all original SAGA extensions and all the ones developed in the previous version of SEXTANTE to gvSIG, using the aforementioned base layer. Some extensions not related with analysis such as input/output ones were, however, not implemented, since they already existed in gvSIG. New ones were also added, up to a total of more than 220, approximately half of which come from the original SAGA version.
3. Including new elements to better exploit the possibilities of the set of analysis extensions. These elements will be reviewed as well in this article.

Currently, SEXTANTE is not a set of extensions for gvSIG, but an independent Java library based on the code developed for that previous gvSIG version. This way it can be easily incorporated into gvSIG to add the same functionalities as the previous version, but also into other GIS applications. This includes other desktop GIS (it is already integrated into OpenJUMP, and prototypes exist for uDig, Kosmo and OrbisGIS), other kinds of applications and libraries, or even to serve SEXTANTE geoalgorithms via WPS (integration with GeoServer and 52North is currently being developed along with the development teams of those projects) Let's see a bit more about SEXTANTE's architecture and philosophy.

The Architecture of SEXTANTE

There are two main parts in SEXTANTE:

A set of base classes which constitute a robust analysis platform and a set of 220+ algorithms built on top of them. These can be used by programmers to incorporate geoanalysis capabilities into their software, simply calling the corresponding algorithm. The design of the base classes makes it easy to call SEXTANTE algorithms regardless which of the data models is used in the application, by wrapping its data objects. Binding for popular libraries such as GeoTools, used in many GIS applications, have already been developed, so in most cases the algorithms can be directly used without having to develop those wrapper classes. Usage of SEXTANTE algorithms from Java code is not covered in this article. Developers can check the SEXTANTE website, where information is given about how to use them or develop new ones based on the core classes of the library.

A set of graphical elements (a toolbox, a graphical model builder, a command line interface...) that make it easier to use and call SEXTANTE algorithms from a GUI. These elements give access to a set of functionalities based on all those algorithms, and they will be described in detail in other sections of this article. All these elements can be easily integrated into a GIS application, adding all the powerful analysis capabilities of SEXTANTE to it.

Having a fixed set of graphical elements also makes it easier for users to adapt to a different desktop GIS, since all operations based on SEXTANTE algorithms will have the same interface. In other words, performing analysis in SEXTANTE is exactly the same whether you use the gvSIG-based version, the OpenJUMP-based one, or any other that might be developed, as long as they reuse the graphical elements provided by the SEXTANTE library.

Of course, developers can create their own graphical interfaces and run algorithms from them, using just the core classes and the algorithms from the library.

Graphical Elements

These are the four main graphical elements of SEXTANTE:

The **SEXTANTE toolbox** (Figure 1), which represents the main element of SEXTANTE. Most users will use only this element in a normal session. From its window, extensions can be run as single processes, and also as batch processes, executing the corresponding algorithm on a set of input data files.

The **Graphical Modeller**. Extensions can be used to define a global process than involve several single processes, each of them consisting of a geoalgorithm. Relation between those processes can be defined so the input of one of them can be the output of a previous one, thus setting a workflow. All this is done through a intuitive interface that we will see shortly.

The **Command Line Interface**. A built in command line for advanced users, which gives more flexibility and allows for the creation of small scripts

The **Command History Manager**. Whenever a SEXTANTE extension is run, a new element is added to the SEXTANTE history. Using this element, the command history can be browsed and certain actions can be repeated, just double clicking on a single command or selecting a block of them (Figure 2).

A toolbar with five buttons gives access to all these elements [4] (Figure 3).

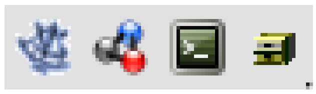


Figure 3: SEXTANTE toolbar buttons

Executing a single algorithm

Running a single algorithm is as easy as double clicking on its name in the SEXTANTE toolbox. A new window will appear, which is automatically generated based on the requirements of the algorithm you are going to execute. However, it is easy to define a different interface if needed, as is shown in Figure 4.

A tab named *Parameters* allows the user to select or enter the input data (layers, tables, numerical values, strings...). New layers and tables are saved by default to a temporary folder. The user can enter a filepath in case he wants to store an output object permanently.

When the algorithm generates raster layers, a second tab named *Raster output* (Figure 5) is found as well. Using this tab, the user can set the extent and

[4] Depending on the implementation, access to SEXTANTE elements can also be done via menu items

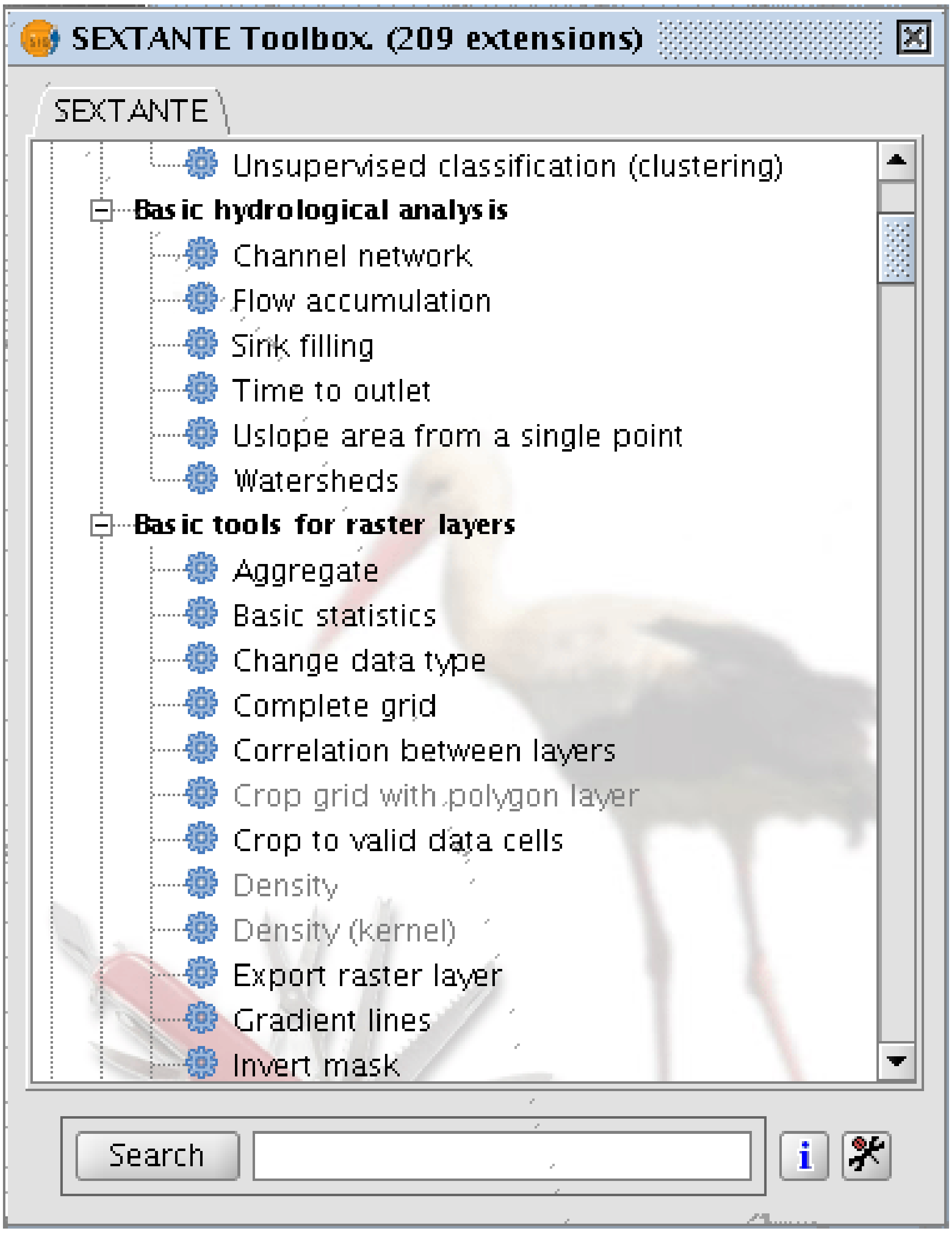


Figure 1: SEXTANTE toolbox

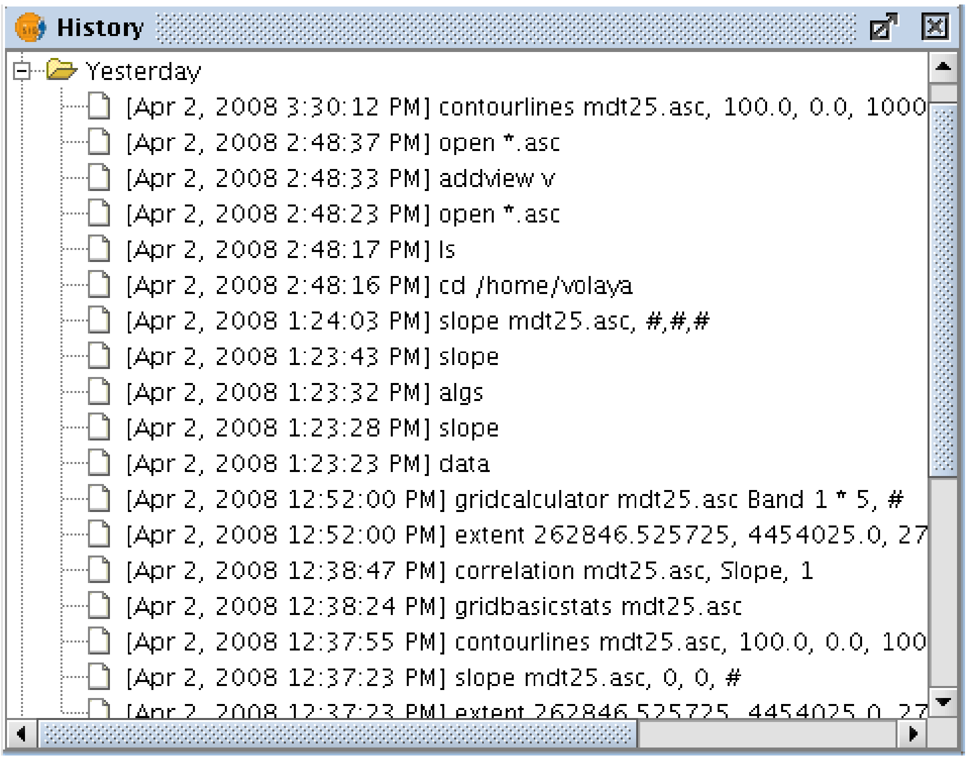


Figure 2: SEXTANTE history

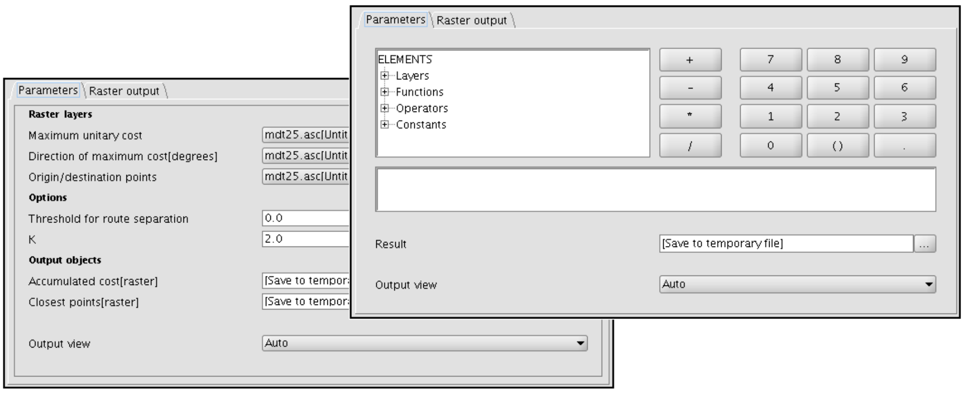


Figure 4: SEXTANTE calculator

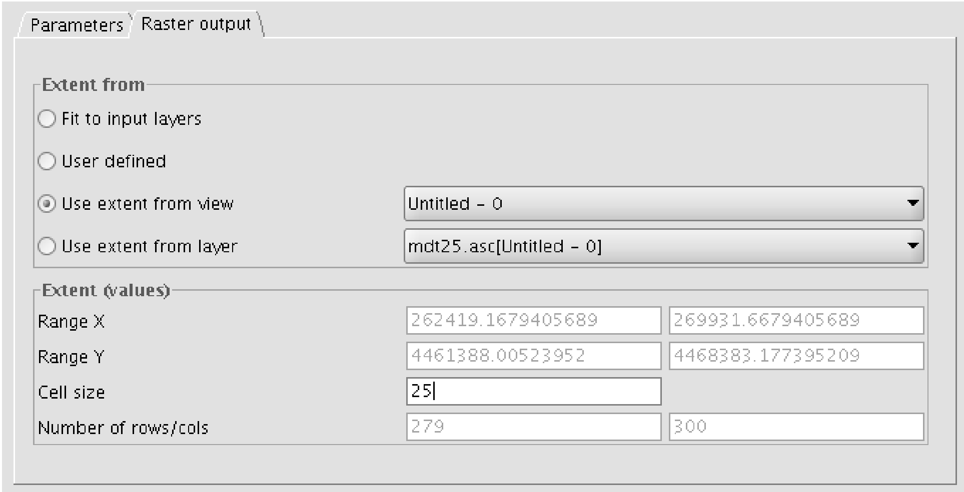


Figure 5: Raster output

cell size for all raster layers generated by the algorithm.

Most GIS which include raster analysis capabilities need layers to have the same raster output and cellsize in order to combine them. Some GIS, like GRASS, require the user to define an output region. In SEXTANTE, this region is defined each time the extension is executed, and the user can enter the corresponding values manually, select them from an existing layer, select them from the extent of a view, or let SEXTANTE adjust it automatically based on the input layers characteristics. This last method is the one selected by default, so most of the time there is no need to use the raster output tab, since the default behaviour is the same as one could expect from any other GIS (specially if there is a single input layer, since the output layers will have its same characteristics).

With this philosophy, data from different sources can be seamlessly integrated to run a process, and the user does not have to worry about preparing those data beforehand. Moreover, preparation of data involves resampling techniques that most users really do not understand enough, being an error prone procedure. Since those resampling tasks are carried out by SEXTANTE, the developer of each algorithm is responsible for how resampling methods are used, so he can let the user select which of them to use or, in most cases, hardcoding it to better fit the algorithm.

Context help for each extension is available, and can be accessed from the dialog box. Users can edit these help files to enhance them or add comments, using the built-in authoring tools (Figure 6). These tools guarantee that help files have the right structure and semantics to be used in different contexts within SEXTANTE, such as the command-line interface that we will later see. Context help can be accessed also directly from the SEXTANTE toolbox, and the whole set of algorithms can be filtered using keywords, so as to make it easier to find the right extension in each case.

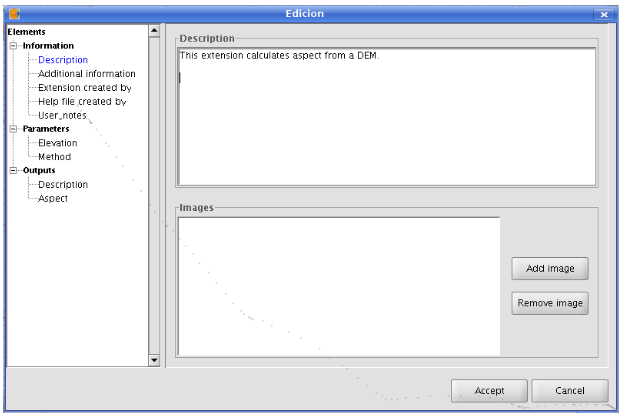


Figure 6: Help system

Creating a Model

When working with a GIS, it is frequent to perform calculations that involve several steps. For example, the Topographic Wetness Index is defined by [1] as

$$\mathrm{TWI} = \ln \frac{a'}{\tan \beta} \quad (1)$$

where a' is the upslope contributing area and β is the slope [1]. Therefore calculating this index in a GIS implies calculating a slope layer from a DEM, and upslope contributing area also from a DEM, and then combining both of them. A great improvement in productivity would be obtained if a user could easily and graphically define a single process that included all these steps, and execute them all at once. While proprietary GIS contain such tools (e.g. *ModelBuilder* in ESRI ArcGIS or *MacroModeller* in IDRISI) no free GIS has a similar functionality. GRASS commands can be used to create scripts, and define compound processes, but most users find it not user-friendly, and would prefer a graphical interface, Attempts have been made to integrate GRASS with scientific workflow systems such as Kepler (see [2], but the complexity of the resulting solution is way beyond the skills of an average (or even advanced) GIS user. SEXTANTE includes a graphical modeller which is unique in the FOSS4G scene, allowing users to quickly and effectively create a model, just following two steps:

1. Defining the inputs needed by the model.
2. Defining the processes to run. Data for each process is introduced through a window similar to the one shown when executing the corresponding extension as a single process. In this case, however, instead of selecting input data from the current gvSIG projects, they are selected from the model inputs (defined in the previous step) and the outputs generated by other processes, thus defining a global workflow.

Figure 7 shows the main window of the model builder, with an example model. On the right side of the window, two tabs are found: *Inputs and Processes*. Double clicking on their elements, these can be added to the model to define its structure. Models are saved in XML-based files, and can be easily shared between users. Once it has been created, the model is treated just as another SEXTANTE algorithm, as if it had been developed directly by programming it. It can be run straight from the model builder or incorporated to the algorithm tree of the toolbox (just saving it to a user-defined models folder), and executed from there as a single process or a batch process, as we will see next. Models can also have their own context help, which can be edited using the help authoring tools of SEXTANTE.

Executing an Extension as a batch process

All SEXTANTE algorithms (including models) may be executed as batch processes. That is, they can be executed repeatedly on a group of entry parameters without the need of calling each time the corresponding extension through the extension manager. This can be used, among other things, to execute one operation (for example, the application of a filter) on a group of layers, like, for instance, all the ones in a given folder. Executing a batch process is not really different from executing a SEXTANTE algorithms in the usual way. The user just has to set the parameters needed to run the corresponding algorithm, and its inputs and outputs. These tasks are done using a table similar to the one shown in Figure 9.

Each line of the table represents an individual execution of the algorithms and the cells of this line contain the values of the parameters, in the same way as they would have been introduced in the different fields of the usual parameters panel. Inputs are not taken from the GIS directly, but directly from data sources (files). Output layers are not added to the GIS interface, but just saved to the selected filepath. In order to make it easier to process large sets of files, several additional functionalities have been added, such as automatic completion of output filepaths using a predefined schema (an enumeration, or the value of an input field) and intelligent copy-paste features). Parameter sets can be saved as comma-separated values, and later opened.

Using the Command-Line Interface

Although most users will prefer using the GUI, the command-line interface provides a powerful way of running SEXTANTE extensions (Figure 9). When a certain task is repeated frequently, it is easier and more productive to use the command-line interface instead of the toolbox.

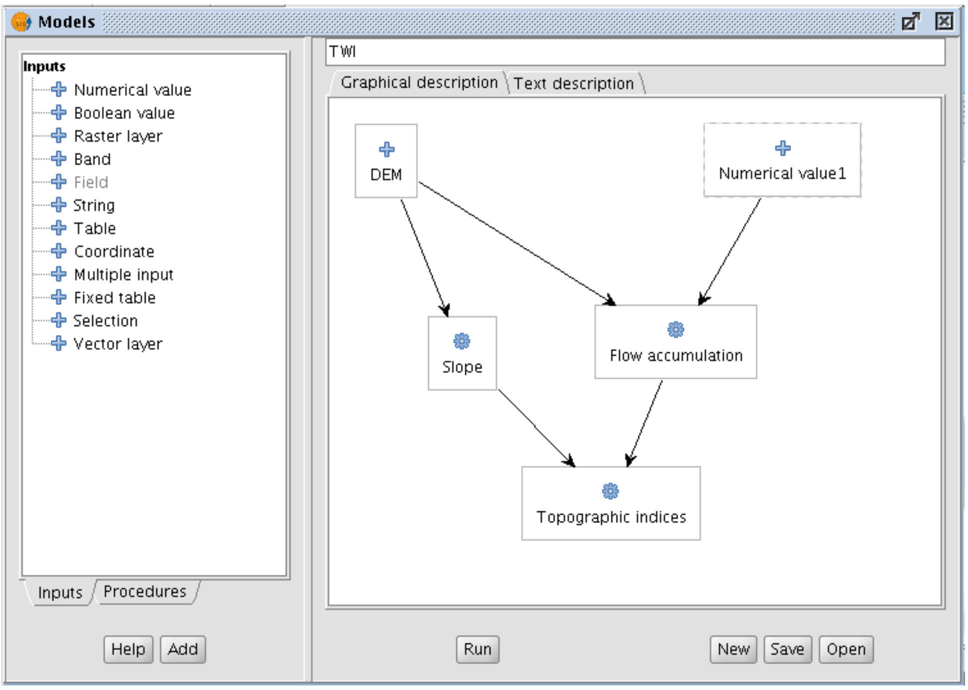


Figure 7: Model builder

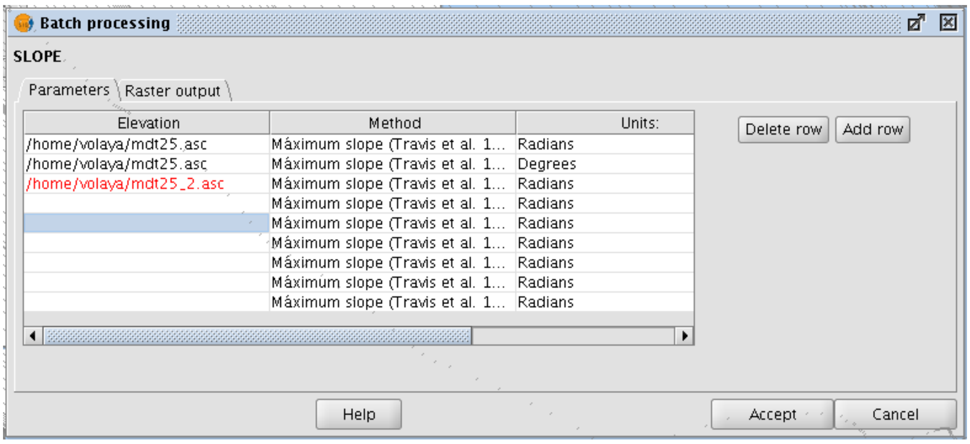


Figure 8: Batch processing

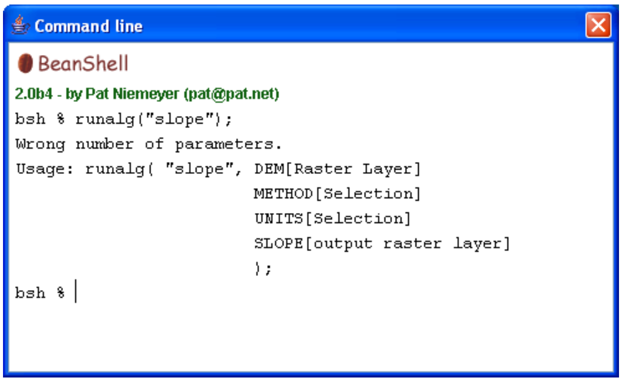


Figure 9: Bean shell interface

Scripts can be written and run from the SEXTANTE interface. They can incorporate standard Java elements such as variables, conditional sentences or loops, since the command-line interface is based on the free Java interpreter BeanShell. Additional commands have been added to allow the execution of SEXTANTE geoalgorithms from it. Scripts can be used to create models and describe compound processes, but there is not yet a link between scripts and models created using the graphical model builder.

The Community

Although SEXTANTE is a small project (three people with just one developer), the community built around it is considerably larger, mainly due the increasing popularity of gvSIG and the effort made by the gvSIG team to promote SEXTANTE along with their own software. Changing from a gvSIG-based conception to an independent library has remarkably increased the number of developers using SEXTANTE. Despite having a large community of users, there was not a large developers community before, something that is gradually changing as more applications start using SEXTANTE as their source of geospatial algorithms. Developers can access SEXTANTE source code using our SVN repository[5], to get a daily updated version.

More Information

For further information, please visit our website at http://www.sextantegis.com or send an email to volaya AT unex.es with your questions.

Victor Olaya
volaya AT unex.es
http://www.setantegis.com

Bibliography

[1] K.J. Beven and M.J. Kirkby. A physically based, variable contributing area model of basin hydrology. *Hydrol. Sci. Bull.*, 24, 1979.

[2] J. Zhang, D. Pennington, and W.K. Michener. Automatic transformation from geospatial conceptual workflow to executable workflow using grass gis command line modules in kepler. *First International Workshop on Workflow Systems*, 3993, 2006.

[5]see https://svn.forge.osor.eu/svn/sextante

Peer-review Papers

Usability Trumps Features

User needs and the redesign of a web-based GIS to support community environmental monitoring

Martin J. Bunch and Michael D. MacLennon

Abstract

Web-distributed tools that complement community-based environmental monitoring (CBEM) initiatives can improve processing of and access to information, supporting environmental education and better informing decision-making. To this end a web-based geographic information system known as 'Juturna' was developed to support CBEM in the vicinity of Toronto, Canada. This web-GIS facilitates input, analysis, and reporting of community data. However, use of the system steadily declined in activity since this initiative started in 2004. Lay users reported that the system was complicated and confusing, and so discouraged use. Also, it employed expensive proprietary software, which was a disincentive for the local Conservation Authority and collaborating NGO to adopt the system.

To revitalize use of the website and provide support to the CBEM program, we undertook to redesign the web-GIS using open source software. To understand why the original web-GIS was not well used and to inform redesign of the system, we implemented a user-centered design methodology. Methods included user testing, rapid prototyping and stakeholder interviews. The process was invaluable in prioritizing user tasks, defining characteristics of users of the website, and identifying those components of the web-GIS most confounding to them. Findings were used to inform re-development of the web-GIS through an iterative process that led to the creation of two prototypes that were evaluated by the user audience and so informed the design of a new (more accessible) website.

Keywords: web-GIS, public participation GIS, user-centered design, iterative development, community based environmental monitoring.

Introduction

Community-based environmental monitoring (CBEM) is becoming an important resource for many environmental monitoring programs. For Conservation Authorities these community-based programs help offset the cost of employing trained scientific experts, and allow members of the public to re-connect with the natural environment as stewards. Creating online tools and resources that complement these community-based initiatives can help improve access to information and allow for more informed decisions to be made on issues that affect the health of these monitored ecosystems.

Through one such CBEM initiative, the Toronto and Region Conservation Authority (TRCA) in collaboration with Citizens' Environment Watch (CEW) invited members of the public to participate in the collection of water monitoring data in Toronto's watersheds. Part of this community-based monitoring program was the implementation of a geographic information system, accessible via the world wide web. This system (known as 'Juturna' after the Roman goddess of wells, springs and streams) facilitates the input, analysis, and reporting of community data. However, use of the web-GIS steadily declined in activity since this initiative started in 2004, while at the same time public interest in the community monitoring program remained strong. Lay users reported that the system was complicated and confusing, and so discouraged use. Also, the web-GIS employed expensive proprietary software, which was a disincentive for the conservation authority and collaborating NGO to ultimately adopt the system.

In an effort to revitalize use of the website and provide support to the community-based environmental monitoring program, we undertook to understand why a seemingly excellent and functional tool was not well used by its intended audience, and then to redesign the web-GIS using open source software. In this paper we briefly introduce the community monitoring program so as to present their information and analysis requirements, and we discuss the application of user-centered design (UCD) methodology as a means of understanding the failure of the initial pilot and to guide redevelopment of the web-GIS to make it more usable to its audience. Specifically, we set out to understand:

- What are the business goals and the website functions most important to the stakeholders?
- How can the Juturna 1.0 system be improved with respect to ease of use its overall usefulness to its audience?

This paper presents a case study in which the pursuit of these questions informed the redevelopment of a web-based public participation GIS. While many professionals in the field may take some of the lessons we present for granted, we offer this case study for those not having system design experience and/or training, in the hopes of saving them missed opportunity, time and resources.

Community-based environmental monitoring and the 'Juturna' web-GIS

Environmental monitoring and the gathering of environmental information are important exercises for understanding the impact of human stressors on the environment. In Ontario, Canada such activities are generally carried out by Conservation Authorities, quasi-governmental organizations that act as stewards to ensure that environmentally sensitive areas remain intact and undisturbed by these stresses. However, reductions in funds for environmental monitoring activities that are allocated by provincial and federal governments has led to increasing examples of local communities partnering with conservation groups so as to continue monitoring ecologically sensitive areas [9] and as a means of empowering and educating members of the public. The regional watershed monitoring program (RWMP) conceived by the Toronto and Region Conservation Authority (TRCA) is one such initiative that implements a collaborative approach to monitoring Toronto's watersheds. Part of this approach was the development of a community-based water monitoring program that allowed members of the community to take part in monitoring the health of the streams and rivers in the Toronto region (Figure 1).

Figure 1: One of the authors (Martin Bunch, right) participating as a volunteer monitor to collect benthic macroinvertebrate samples on a transect in Black Creek near York University, Toronto.

As part of the RWMP the Conservation Authority facilitated annual workshops to train members of the public as volunteers for gathering biological and abiotic data from streams, and to identify the benthic macroinvertebrate (BMI) organisms (mostly bugs and worms but also potentially clams, snails and crayfish) captured in their samples. The purpose of these workshops was not only to educate members of the public about water monitoring and the watershed ecosystem but also to ensure that data is collected as accurately as possibly. TRCA, in cooperation with the environmental NGO Citizen's Environment Watch developed monitoring protocols that use benthic macro-invertebrates and biotic indicators to measure aquatic health. CEW committed to provide education, equipment, and support for community-based ecological monitoring work, to teach annual workshops, and to operate the volunteer monitoring program.

Community volunteers who participated in the program generated data that reported the frequency of various organisms in their samples, geomorphic and vegetative characteristics of monitoring stations, and the station location. This information can be used to construct metrics that indicate whether a stream's health is unimpaired, potentially impaired or impaired, based on the frequency distribution of the organisms collected and their known sensitivities to stressors [1, 4, 5]. TRCA staff involved in BMI monitoring would also generate water chemistry data. While TRCA scientists are practiced at managing and analysing this data, the lay volunteers participating in the monitoring program generally are not.

To address this, TRCA enlisted researchers (our predecessors) at York University's Faculty of Environmental Studies to develop a system to assist in the uploading, storing, reporting and analysis of volunteer data, as well as data generated by TRCA staff. This resulted in the creation of the web-GIS called 'Juturna.' The website itself also served other purposes for different individuals with a variety of interests in environmental monitoring. For instance, the general public could use the website to access TRCA's aquatic monitoring data to perform rapid analysis of watershed and subwatershed data.

In October 2004 the pilot project was implemented. It was one of the first community-based environmental monitoring projects to utilize the internet and rely on a complex web-based Geographic Information System for interacting with the water monitoring data [2]. The web-GIS allowed users to find the location of their monitoring station using postal code searches, aerial photos, street networks, surface water layers, land use, and points of interest. They could then create a point location with which to associate the monitoring station data, upload data and generate reports that included calculation of Benthic indices (indicating stream health).

The pilot project was met with great interest throughout Toronto's environmental conservation community and the initiative was presented at several conferences. The annual water monitoring workshops that were used to train new volunteer water monitors were always filled to capacity. Coming into this project (Martin Bunch inherited the project as a faculty member, and Michael MacLennan was a graduate student at the time), we were very impressed with the system. It was highly functional, allowing users to upload everything from their BMI data to digital photos of their monitoring stations (which were incorporated into the station reports) (Figure 2). It produced extremely descriptive and well organized reports at the monitoring station, subwatershed and watershed level, it had an attractive web interface, sophisticated administrative tools, and most of all it worked consistently and well.

However, we were surprised to discover that there was a steady decline in the number of visitors to the website, despite the ongoing interest in the water monitoring workshops by members of the public. More disturbing was that in the first monitoring season (2005) supported by Juturna, data for only 13 volunteer monitoring stations were uploaded – less than half of what was expected. This suggested that Juturna 1.0 was not effectively supporting the needs of volunteer users. Furthermore, informal feedback suggested that users may have become frustrated with the data upload process and abandoned the process before completing it.

As we looked into modifying the system to deal with this, two other issues arose. First, software needed to be updated and data made current. Our University had licences for the proprietary software to support the update (ArcIMS, ArcSDE Crystal Reports, MS Windows Server) and also access to the needed data (digital air photos, vector layers for roads, surface waters, watersheds, parks and other features) except for proprietary postal code data. However this meant that the program was dependant on the University for software maintenance and support and on the University and/or the TRCA for data updates. Purchasing both the software and data licensing was not feasible for an NGO like CEW whose annual budget for this project was quite small. Second, the back-end systems of this extremely complicated application indicated that maintenance was going to be difficult. There was only very sparse documentation on system architecture and design, and the code was not commented. It was obvious that the design of the original system did not consider maintenance or ongoing development (URLs, for example, were hard coded in the system).

In the winter of 2005–2006, it became clear that the website needed to be redeveloped to alleviate these problems. In early spring of 2006, financial support was provided by the Ontario Ministry of Environment to redevelop the website using Free and Open-Source (FOSS) software. The primary open source applications we used were MapServer, PostgreSQL/PostGIS on a Linux platform with SQL, Ruby, HTML, and Java components. The open

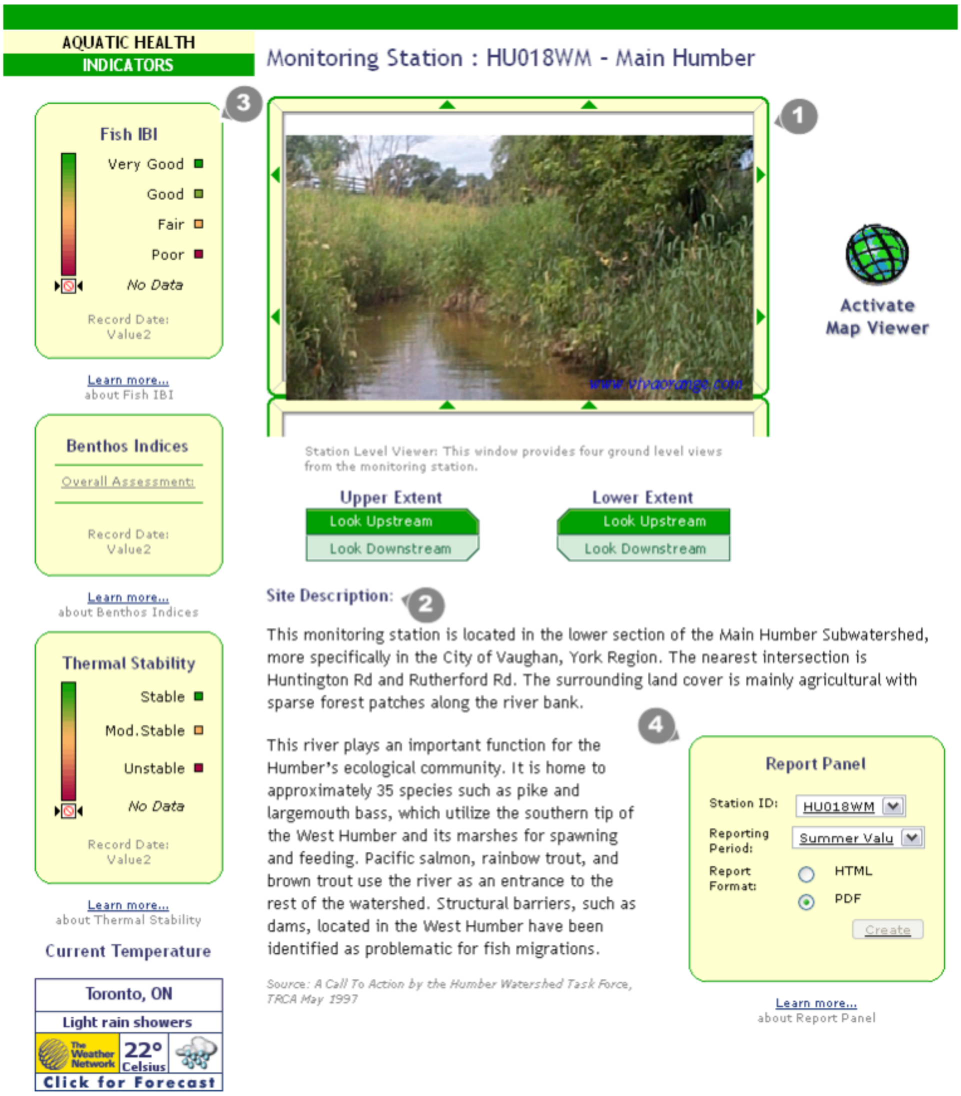


Figure 2: A typical water monitoring station web page in Juturna 1.0, including 1) digital photos of the monitoring station, 2) site description, 3) health indicators and 4) a report generation panel.

source approach provided us with the opportunity to develop the site in a user-centered design paradigm with the users at forefront of design decisions.

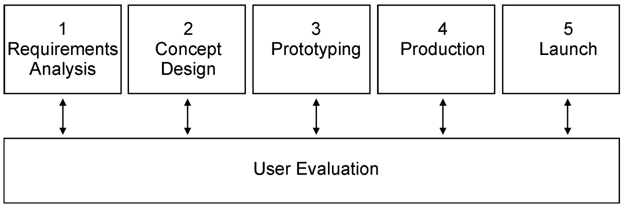


Figure 3: Pervasive Usability process (after [3]).

User-Centered Design and Iterative Development

Good design of any product (Internet-based, computer based or otherwise) is most successful if the design group understands as much as they can about the users' intended use of the product [3, 7, 8]. User-centered design (UCD) methodology takes this to heart. It is an approach to software development and web design that allows for a deeper understanding of the user. Having observed that there were serious usability problems with Juturna 1.0, we undertook to implement a user-centered design methodology in the re-development process of the Juturna website.

The UCD procedure consisted of a series of development stages that incorporated information gathered from the implementation of several usability methods. [7] defines this kind of approach as an iterative development framework. The idea of iterative development (ID) is founded on the premise of continual refinement through a trial and error process whereby each successive iteration incorporates what is learned from the applied usability methods. The end goal of the project being a product (the Juturna 2.0 web-GIS) that is as usable as possible, meeting the needs of its intended audience.

Usability can be measured on three factors; functionality, desirability and efficiency [7]. Determining how these are achieved depends on how well the development team understands its user audience. In order to do this [3] suggest adopting a 'pervasive usability' approach within an iterative design process. The purpose of this approach is to employ a methodology in which user input is incorporated in every step of the development process. For example, when a change is made to the design of the interface it is tested by users to determine if this change was effective in achieving a usability benchmark. This allows for different usability methods to be implemented at each stage of the development process (Figure 3). In this project the majority of the usability tools were implemented in the first three phases of the project. These tools include stakeholder interviews, use cases, user flows, and user testing.

Methods and the development process

Following the process outlined by [3] and [7] our design process was implemented in five stages: requirements analysis, conceptual design, mockups and prototypes, production and launch. These stages were operated by the application of the techniques presented in Table 1. The process was used both to evaluate the Juturna 1.0 system (to understand its failure), as well as to inform redevelopment of the new open source system.

Stage One: Requirements Analysis

The first stage in the redevelopment of Juturna identified the goals and parameters of the project. Using stakeholder interviews, a baseline evaluation of the existing website in terms of its current functionality as well as its usability was performed. This was followed by stakeholder interviews to explore the purpose of the site from a business perspective, understand the stakeholder organizations' perspectives of the users, and to set the context for the website's future direction. These interviews contributed both to evaluation of the existing Juturna 1.0 system and to determining the target audience, user goals, business goals and technical requirements for the redeveloped website. At this stage user needs and target usability goals were determined as a means of guiding the following stages. This process identified five user groups: volunteers, TRCA staff, system administrators, consultants and partners, and the lay public.

Stage Two: Conceptual design

Analysis and evaluation of the Juturna 1.0 site against data derived from stakeholder interviews was then used to define a concept for the architecture of the initial design of Juturna 2.0. This analysis allowed us to produce flowcharts that summarized how the major user tasks were to be completed in the new system (e.g., Figure 4). These documents

Technique	Description
Semi-structured interviews	Semi-structured Interviews (SSI) occurred over a one hour period in a comfortable setting chosen by the interviewee. A limited set of open-ended questions were used to direct a discussion. More specific questions were asked depending on the responses of the interviewee. Themes explored in the SSI included, for example, history and context of the organization with respect to water monitoring, and the efficacy of the Juturna 1.0 system.
Use cases	User scenarios were developed for each of the identified user groups. Use Case Scenarios are narratives about how a particular type of user of the system would interact with it. These narratives provide some context and "feel" for how the system needs to work. From these scenarios a functional requirements list and user flows may be constructed.
Task analysis	Tasks are expressed as simple flow charts ("user flows") that describe the possible flow of interaction of particular category of user with the website. These assist in developing the information architecture of the system, that is, the design and organization of the navigation structure and consideration of position and placement priority of page components like the log-in form, web-GIS functions, benthic data entry forms and data access rules.
Wireframe mockups	Wireframe mockups are rapid prototype application interfaces. This is a block diagram that illustrates the overall navigation of a website and the blocks of elements (such as content and functionality) that will be present on the screen.
Heuristic evaluation	Heuristic evaluation systematically assesses a user interface design for usability in form and function based on a set of rules. In this case, rules were derived from Nielsen's 10 usability heuristics: visibility of system status; match between system and the real world; user control and freedom; consistency and standards; error prevention; recognition rather than recall; flexibility and efficiency of use; aesthetic and minimalist design; help users recognize, diagnose, and recover from errors; help and documentation [11].
User evaluation	For the original Juturna 1.0, for prototypes, and for the re-designed open source system, user testing involving various user groups was undertaken. Five evaluators were used to satisfy the minimum number to catch eighty percent of the usability problems, which is between four and five ([10];[13]. Six tasks exploring four main functionalities were evaluated by participants. These were 60 minute interviews in which evaluators undertook tasks on the system, while the moderator took notes and recorded quantitative metrics about speed of completion and ease of use. Users, sitting at a computer accessing Juturna, responded to questions such as "You would like to generate a report of the Humber watershed for the 2001 year. How would you generate this report?" by undertaking the task on the system, and vocalizing their thoughts as they did so.

Table 1: Primary techniques employed in user-centered design of Juturna 2.0

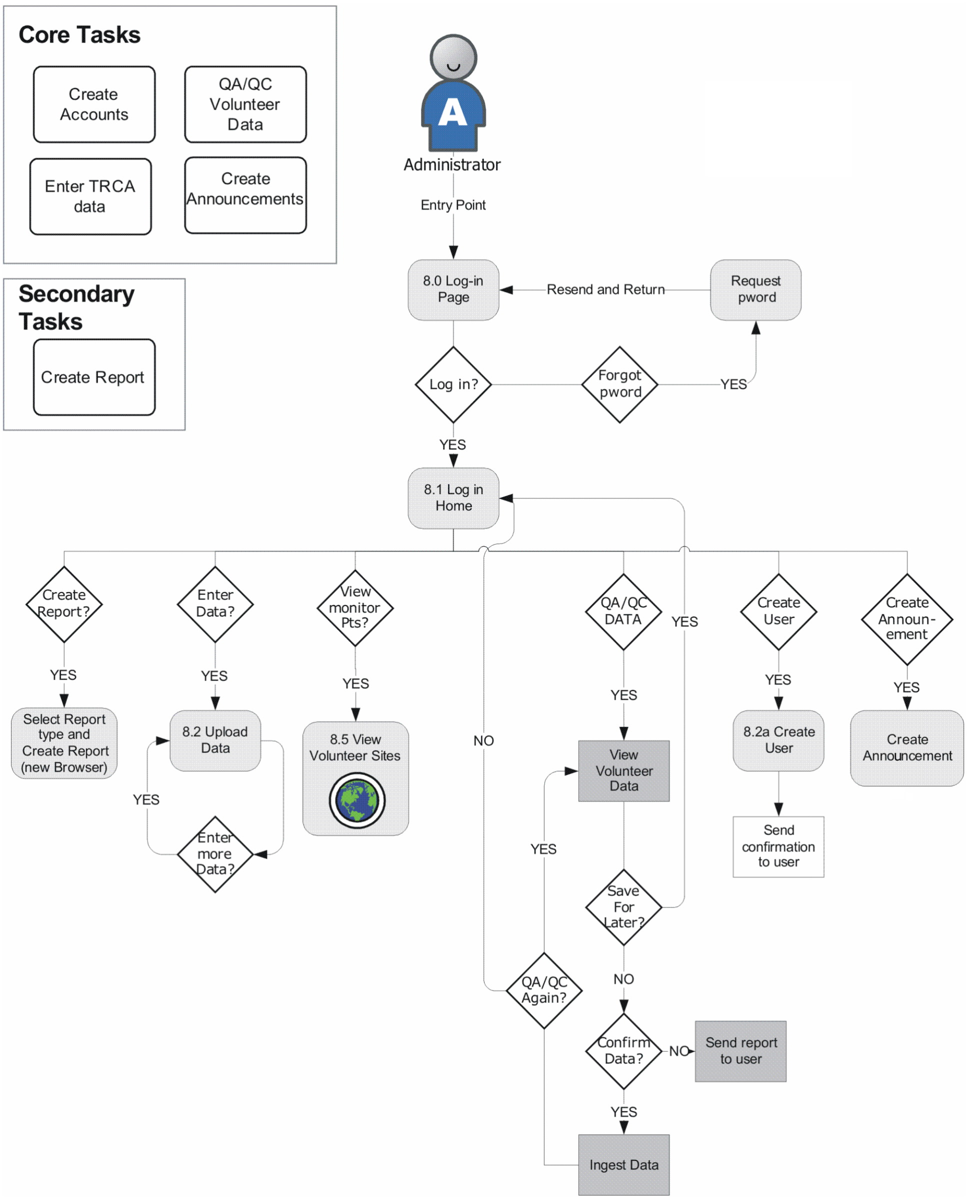


Figure 4: 'User flow' describing possible flows of interaction of the administrator user with the website.

provided a means of sketching the website's organization in a more formal technical document known as a 'use case' that defined user scenarios. These scenarios were developed based on interviews with the major stakeholders and our previous experience with the Juturna 1.0 website.

Stage Three: Mockups and Prototypes

An interim representation of the new website was implemented using mockups and prototypes. This allowed for early evaluation of the new design before the final system was produced. Rapidly producing a prototype of the final website using simple diagrams allows for changes to be implemented easily and quickly, preventing extensive modifications from occurring later in the development process. Figure 5 presents a wireframe prototype used for this purpose. The wireframes lacked content but had operational navigational structure. User testing of these mockups, and consultation with the stakeholders, identified changes to be implemented in the production stage.

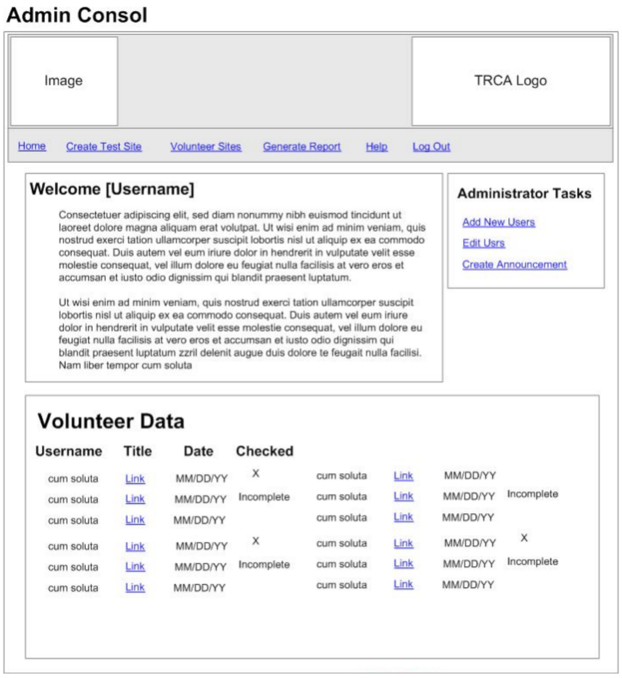


Figure 5: Wireframe prototype of the administrator console.

Stage Four: Production

At the production stage the product was created. Content was added to the website, graphic elements were developed, and coding was produced for the components of the website. Evaluation at this stage occurred in the form of heuristic evaluation to assess the level of usability in comparison to Juturna 1.0. This was a formal comparison of the website with usability rules, so as to predict usability of the system [6, 12]). User testing of the website also occurred to ensure that the website functioned properly.

Stage Five: Launch

At this stage the product was finally launched and made available to the public. Prior to making the website available, a quality assurance test was performed to ensure that all necessary website functions were working properly. This was not an endpoint as the system continues to be refined as further development phases are implemented, new functionality is added, and user needs change.

Assessing Juturna 1.0 and building Juturna 2.0

User testing occurred using the original Juturna 1.0 system (so as to provide insight into its failure and to inform development of the new open source version of Juturna), on a wireframe mockup of the new system, and on a redeveloped version of the system. Primary functions evaluated included: data upload, data download, report creation, user account management, and data assessment functions. These functions were chosen because they were determined through the UCD process as important to the website for all user groups. User groups identified were: Volunteer users (community monitors), consultant users or partners (who may need to access raw data in the system), Administrator users, TRCA users (staff), and Lay users (the general public). Volunteer users and TRCA users were the primary users of the site, and their activity encompassed that of the others. Thus, user testing was undertaken on the basis of these two groups. This process led to four main areas of improvement in the system relating to: task efficiency, topology of navigation structure, reducing users' memory load and bullet proofing against user error.

Improving task efficiency

We were able to improve the efficiency of several tasks in the redevelopment of the website. For example, the action of logging into the website is one of the most frequently performed activities by all but the lay user group. Juturna 1.0 placed the log in function several pages deep within the sub-navigation structure of the site, instead of in an apparent location such as the monitoring sites landing page. In redeveloping the site we placed the log in function on the home page to allow users to easily locate the log in console (similar in location to several popular websites like google.com, gmail.com, msn.com and hotmail.com). This simple modification reduced the total time spent completing these tasks and also saves the user from the frustration of having to search the website for the log in page.

Figure 6 illustrates this point in relation to the number of mouse clicks required by the user to log into the restricted portion of the website. The original website required the user to click four different links on four separate web pages before he or she can start uploading data. The redeveloped website reduced this to one mouse click by placing the log in function on the home page of the website, and also placed data uploading functionality as one of the first tasks a user's log in screen.

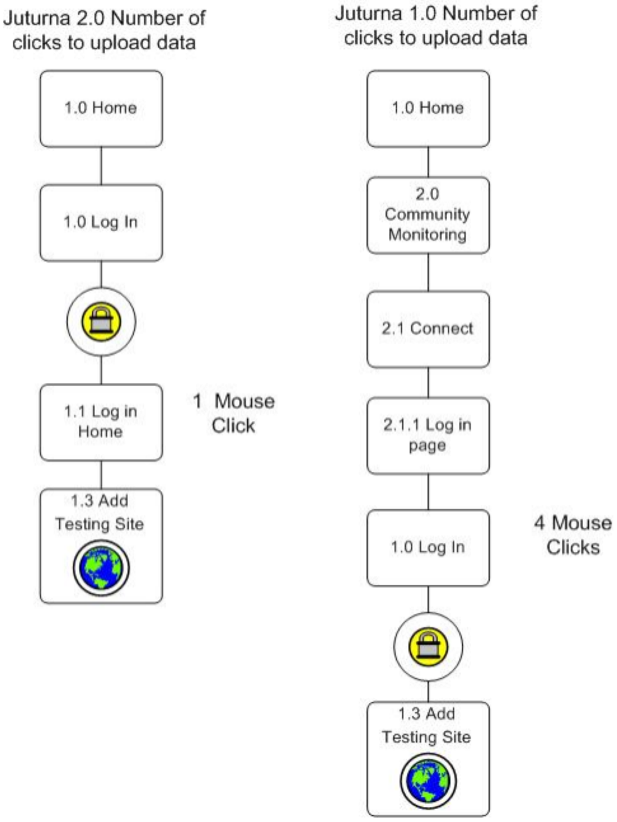


Figure 6: Illustration of click events needed to log in.

The purpose of this approach was to improve the speed at which a user could start a task by reducing the cognitive load required to remember which links to click to get to the data upload screen. Quantitative metrics (Table 2) as well as user vocalization during user-testing interviews was useful in elucidating issues with task efficiency.

Topology of navigation structure

Initial evaluation of the Juturna 1.0 navigation structure found that the navigation was largely inconsistent and arbitrarily defined across the website. For instance, the home page of Juturna 1.0 presented the primary navigation on the left side of the page while all other pages moved the navigation structure to the top of the page. This slight change presents an inconsistent design that has the potential to be highly disorienting to users, giving them the idea that they have moved to another website. To further confuse the design, common names associated with primary navigation on the home page also changed when the user navigated to other pages on the site. For example, the primary navigation on the home page consisted of several links labeled "Administrator Area (secure)", "Community Monitoring" and "Report Generation Area (secure)" while the primary navigation menu found on all other pages labeled these same links as simply "Admin", "Community" and "Report Generator."

Additionally the web pages of Juturna 1.0 tended to be organized in an ordered sequence with certain pages enabled with a sequentially meshed navigation structure. Figure 7 shows the topological structure of Juturna 1.0. Use of an ordered sequence or linear navigation structure works well when users must complete a series of tasks in a specific way (e.g., where a user must upload data in a particular fashion). However, this kind of linear approach is problematic when implemented over an entire website because it prevents the user from exploring a site in a free and open manner, which allows them to decide the order in which information is consumed. The meshed navigation structure implemented in Juturna 2.0 (Figure 8) alleviates this problem by allowing users to navigate to any and every page on the website while still allowing for some areas of the site to adhere to a sequential structure. The downside to this approach is that users have the ability to step in and out of different stages of the linear structure which may break the data uploading process. It is best to monitor for this and communicate this error if a user deviates from the sequential structure.

Task One: Report Generation						
	Craig	Derrick	Samantha	Tanya	Chris	Average
Time to Complete	2	1	2	3	1	1.8
Errors	1	2	2	3	1	1.8
Key	**Time to Complete** 0. Fail 1. Succeed very slowly in a roundabout way 2. Succeed a little slowly 3. Succeed quickly			**Number of Errors** 0. Fail because of errors 1. Many errors 2. Some errors 3. Few or no errors		

Table 2: Task One Quantitative Metrics (names are pseudonyms)

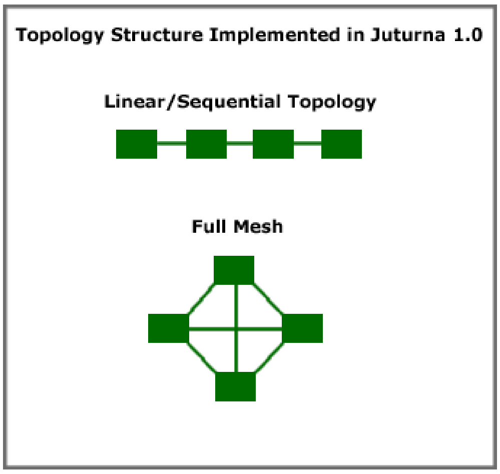


Figure 7: Navigation topology used in Juturna 1.0.

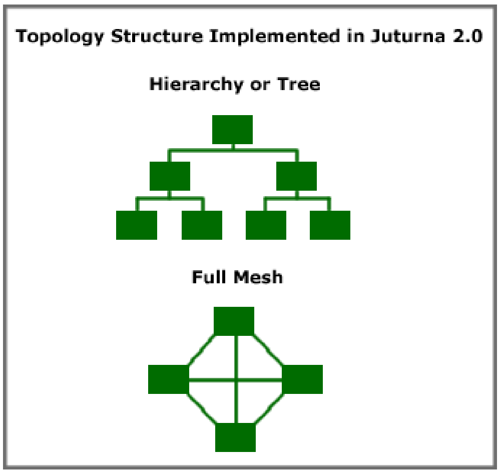


Figure 8: Navigation topology used in Juturna 2.0.

Instead of maintaining a navigation structure that implements both sequential and meshed architectures, we designed the first prototype with a hierarchical and meshed topology that used navigation titles that are task-based. As illustrated in Figure 8, this approach organizes the website around the principle of tasks that a user is able to accomplish on the site allowing them to easily orient themselves and get to the task they want to accomplish. For example, the navigational labels for a user account having permissions to upload data and create reports were changed to 'create test site' and 'generate report' instead of the more general and ambiguous title of 'community monitoring.' Designing the website in this manner created an architecture that will be familiar to the user for two reasons. First, the hierarchical structure is the most commonly used navigation structure on the Internet and is therefore well understood by the user audience. Second, because its organization is task based, there is a greater potential that it appears more intuitive to the user audience because it fits the user's understanding of the purpose of the site.

Reducing users' memory load

Efforts were also made to minimize the user's memory load by making options and links more visible. [3] argue that the user should not have to remember information from one part of the website to another. Instructions for use of the system should also be visible and easily retrievable whenever appropriate. Juturna 1.0 does not minimize the user's memory load because it makes use of a collapsible navigation structure. The system should also always keep users informed about its state. For example, when a user clicks an element on screen, the response from the system should be fast and clearly indicate the navigation path chosen by the user, e.g., by representing visited links by a color change. These are some of the ways websites can inform the user of changes to its state as they interact with the system. Juturna 1.0 did not keep the user informed in an effective manner. Instead it forced users to explore the website through a trial and error process and did not indicate what pages the user has visited. This is particularly relevant in the navigation menu where users must 'discover' what links are available by expanding the hidden menus. Figure 9 shows the Juturna 1.0 navigation menu of the monitoring link as it appeared when the user dragged their mouse over the link. This design reveals a highly complex secondary navigation structure that requires the user to remember the location of sub-navigation items that are always hidden.

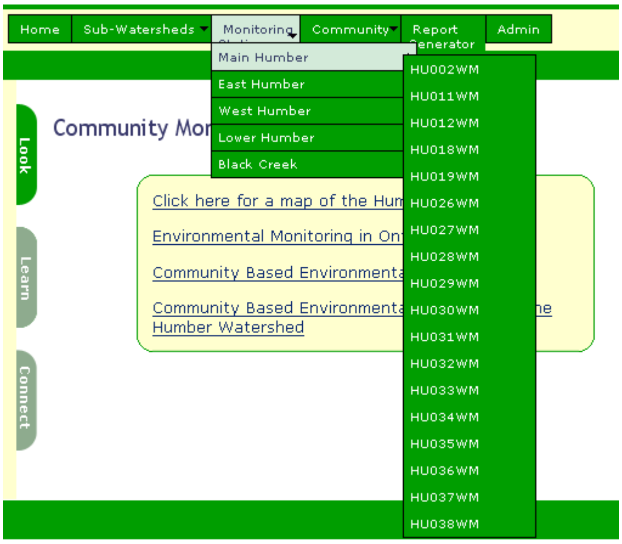


Figure 9: Hidden navigation menu of Juturna 1.0's navigation pane.

Juturna 2.0's interface is more consistent in keeping the user informed and orienting them to the purpose of the website. The navigation menu does not contain any hidden links and there are lots of visual signs in the form of page titles that show the user where they are in the web site. Figure 10 illustrates the title of a web page in Juturna 2.0. Having the title correspond to a link in the navigation bar at left shows the user their location in the navigation structure of the website at all times. This approach reduces the load on the user's memory by utilizing location cues.

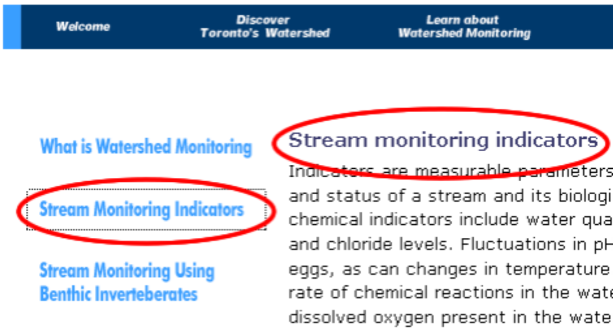


Figure 10: Visual indicators in Juturna 2.0 help users know where they are at all times.

Bullet proofing

It was also determined through the analysis of the Juturna 1.0 system that if improvements were made to the system's ability to prevent and catch user errors the website would be significantly improved. Juturna 1.0, contained many flaws in its ability to catch and prevent user errors in several key functions including data downloading, data uploading and report generation. The inability of the system to catch these errors greatly limited the ability of users to complete tasks. This is particularly apparent in the data downloading function. Initially this function required the user to select several criteria including the geographic extent of their data, the sampling season, the sampling year and the data type, prior to downloading data. However, there was no information present in the selection parameters to let the user know whether they were requesting data that did or did not exist in the database. As a result, the user was faced with the responsibility of discovering what data was present through a trial and error process. The initial redevelopment prototype attempted to alleviate this problem by using a series of selection menus that only listed the parameters present in the database.

Conclusions

Because we were a small team (a professor drawing upon graduate student labour, and contracting out some programming work) with a small amount of funds, the process described above did not demonstrate the rapid response of agile design. However, it did yield responsive development – the new Juturna system is leaps and bounds ahead of the earlier system in terms of usability. Paradoxically, it is also much less impressive. Juturna 2.0 has fewer functions, and is less visually powerful than its predecessor. This is because many of the functions incorporated in the first version were not essential to its mission. Some of those functions and capabilities may return in future iterations of development, but only if users indicate a strong need for them and if they can be implemented without requirements for proprietary software or data.

Currently, Juturna 2.0 resides at `www.juturna.ca` and is a functional web-GIS supporting community volunteer monitoring organized by Citizens' Environment Watch. After the first redevelopment phase of the system was completed in late 2007 there was a delay in implementing Juturna (though it was online and functional) due to a reorganization of the monitoring program by CEW and changing in-house needs at TRCA due collaborative data management initiatives with other CAs. This necessitated another iteration of the design process in 2008. The system is now being populated with volunteer data from the 2009 season and will support a new orientation of the CEW program to operate with schools in the Toronto region in 2010. Future plans for Juturna target interoperability, as well as modifications to make the system easily adaptable to other watersheds outside of the TRCA jurisdiction. Once this is happens we intend to make the system available in a free and opensource manner.

Acknowledgements

The authors wish to acknowledge the contributions and collaboration of our partners at the TRCA and Citizens' Environment Watch, as well as Anna Goldberg and Joseph Zhao at Malone Given Parsons Limited for programming work, and Matt Milan for advice in the redevelopment of Juturna 2.0. This project has received funding support from the Ontario Ministry of the Environment. Such support does not indicate endorsement by the Ministry of the contents of this material.

Martin J. Bunch
Faculty of Environmental Studies
York University
4700 Keele Street
Toronto ON Canada
M3J 1P3

Tel: 416-736-2100 ext: 22630
Fax: 416-736-5679
Email: bunchmj@yorku.ca

Bibliography

[1] M. T. Barbour, J. Gerritsen, B. D. Snyder, and J. B. Stribling. *Rapid bioassessment protocols for use in streams and wadeable rivers: periphyton, benthic macroinvertebrates and fish, second edition.* Washington D.C.: United States Environmental Protection Agency, Office of Water., 1999.

[2] J. Borisko. Introducing Juturna: A Data Management and Mapping system for Greater Toronto. *The EMAN Monitor (online)*, 2(1), 2004.

[3] T. Brinck, D. Gergle, and S. D. Wood. *Designing Web Sites That Work: Usability for the Web.* San Diego, California: Academic Press., 2002.

[4] J. Castela, V. Ferreira, and M. A. S. Graca. Evaluation of stream ecological integrity using litter decomposition and benthic invertebrates. *Environmental Pollution*, 153(2):440–449, 2008.

[5] W. L. Hilsenhoff. Rapid field assessment of organic pollution with a family-level biotic index. *Journal of the North American Benthological Society*, 7(1):65–68, 1988.

[6] E. T. Hvannberg, E. L. C. Law, and M. K. Larusdottir. Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers*, 19(2):225–240, 2007.

[7] M. Kuniavsky. *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco: Morgan Kaufman Publishers, 2003.

[8] J. Lazar. *Web Usability: A User-Centered Design Approach*. Boston: Addison-Wesley Longman Publishing, 2005.

[9] C. Mayfield, M. Joliat, and D. Cowan. The roles of community networks in environmental monitoring and environmental informatics. *Advances in Environmental Research*, 5:385–393, 2001.

[10] J. Nielsen. Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. Online, 1994, revised 2006. URL http://www.useit.com/papers/guerrilla_hci.html.

[11] J. Nielsen. Ten usability heuristics. Online, 2005. URL http://www.useit.com/papers/heuristic/heuristic_list.html.

[12] J. Nielsen and R. L. Mack. *Usability Inspection Methods*. New York, NY: John Wiley & Sons., 1994.

[13] J. Rubin. *Handbook of Usability Testing: How to Plan, Design and Conduct effective tests*. New York: Wiley, 1994.

Volunteer Recognition

The OSGeo Journal is written, edited, and published by volunteer members of the OSGeo Foundation. This journal would not be possible without the hard work and dedication of our volunteers.

We would like to recognize the hard work of the following volunteers on Volume 6 of the OSGeo, and want to give them our sincere thanks for their efforts.

Daniel Ames

Dr. Daniel P. Ames is an Associate Professor of Geosciences at Idaho State University, Idaho Falls, Idaho, USA. He directs the Geotechnologies graduate program which offers MS and PhD degrees in Geographic Information Science and Engineering and Applied Science, respectively.

His research program is centered on the open source MapWindow GIS project (www.MapWindow.org), integrated environmental modeling (www.iemHUB.org), and hydroinformatics (www.HydroDesktop.org) under funding from the US EPA, NOAA, NASA, DOE, FEMA, and various commercial entities. He can be reached by e-mail at amesdani@isu.edu.

Landon Blake

Landon Blake is a licensed land surveyor interested in the intersection between GIS and land surveying. He is an advoccate of open source software and open technology standards. He is a charter member of the OSGeo and a volunteer programmer and administrator of the JUMP Pilot Project.

Scott Mitchell

Scott Mitchell is co-director of Carleton University's Geomatics and Landscape Ecology Research Laboratory in Ottawa, Canada, and 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, especi ally 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 at glel.carleton.ca.

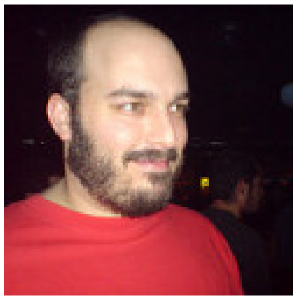
Tyler Mitchell

Tyler Mitchell is the Executive Director of the Open Source Geospatial Foundation (OSGeo) and Editor in Chief of the OSGeo Journal. He serves on OSGeo committees, speaks regularly at events and works closely with members to further the OSGeo mission. His book, Web Mapping Illustrated (O'Reilly, 2005) was one of the first books addressing open source mapping technologies, in particular MapServer, PostGIS and GDAL/OGR.

Barry Rowlingson

Barry Rowlingson is a Research Associate in the School Of Health and Medicine at Lancaster University in the UK. His research is in the field of spatial statistics, mostly applied to disease outbreak modelling and prediction. He has contributed to several open-source projects, most notably the R statistical software system and Quantum GIS.

Jorge Sanz



Jorge Sanz is a consultant in Geospatial technologies at Prodevelop (Valencia, Spain). He received education and training in surveying, cartography and geodesy at the Polytechnic University of Valencia.

Nowadays he is the coordinator for the gvSIG Technical Steering Committee and is in charge of the incubation process of gvSIG at OSGeo. He is a Charter Member of the OSGeo Foundation, a member of the OSGeo Incubation Committee and serves on the Board of Directors of the Spanish language Local Chapter of the OSGeo Foundation.

Micha Silver

Since 1996 Micha Silver has been doing network administration and GIS consulting for local government organizations in the Arava region of Israel, including the Arava Drainage Authority and the Dead Sea Arava Science Center. In day-to-day work he uses a mix of proprietary and open source software.

Zachary Woolard

Zachary Woolard has a B.S. in Applied Geography. He has worked for six years in both government and private sector jobs, doing a little bit of everything along the way. His primary interests are open data standards, OSGeo in local government, and learning everything he can about databases, both open and proprietary.

Imprint

Editor in Chief:
Tyler Mitchell - tmitchell AT osgeo.org

Assistant Editor:
Landon Blake

Section Editors:
Daniel Ames
Scott Mitchell
Barry Rowlingson
Jorge Sanz
Micha Silver
Zachary Woolard

Acknowledgements
Various reviewers & writers

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the LaTeX 2ε style source has been kindly provided by the GRASS and R News editorial boards.

Journal online: http://www.osgeo.org/journal

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

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8

ISSN 1994-1897

OSGeo

Your Open Source Compass