



FOSS4G

D E N V E R 2011

12-16 SEPTEMBER 2011 SHERATON DENVER DOWNTOWN
2011.FOSS4G.ORG

Conference Proceedings

**Free and Open Source Software for
Geospatial Conference**

**OSGeo Journal
Vol. 10 - Feb 2012**



Volume 10 Contents

FOSS4G 2011 Conference Proceedings	2	Using GRASS and R for Landscape Regionalization through PAM Cluster Analysis	26
Editorial - FOSS4G 2011 Academic Chair	3	Functional Coverages	32
Open source based online map sharing to support real-time collaboration	5	Opticks Open Source Remote Sensing and Image Processing Software	45
An innovative Web Processing Services based GIS architecture	15	Implementation, challenges and future directions of integrating services from the GIS and decision science domains	50
		A Vivid Relic Under Rapid Transformation	56

Welcome from the Conference Chair



Welcome to this special edition of the OSGeo Journal, featuring selected papers from the academic track that were presented at the FOSS4G (Free and Open Source Software for Geospatial) 2011 conference in Denver.¹ The conference was the largest FOSS4G yet, with 914 attendees from 42 countries. Feedback from attendees was very positive, with the post-conference survey giving it an overall rating of 4.32 out 5. The attendance reflects the strong growth in interest in open source software that we are currently seeing in the geospatial industry.

We made a conscious effort in 2011 to enhance the academic track at the conference by providing improved publishing opportunities. We did this through publishing papers both in “Transactions in GIS” and in this edition of the OSGeo Journal. I would like to thank Rafael Moreno for leading this effort, as well as the rest of the organizers of the academic track who Rafael recognizes below.

*Peter Batty, Ubisense
FOSS4G 2011 Conference Chair*

¹FOSS4G: <http://foss4g.org>

FOSS4G 2011 Conference Proceedings

Editorial - FOSS4G 2011 Academic Chair

Rafael Moreno
 University of Colorado Denver
 Department of Geography and Environmental Sciences

The growing interest in Free and Open Source Software for Geospatial applications (FOSS4G) requires specialized publication venues.

The FOSS/FOSS4G movement is attracting increasing attention from end users, developers, businesses, governments, educators and researchers around the world (e.g. Weber 2004, CRM-Reviews 2006, Faber 2007, Garbin and Fisher 2010). This attention is in part sparked by (and in turn it demands) more access to information about FOSS4G and its applications. Specialized high quality publication venues like the OSGeo Journal are critical to providing access to the latest developments, to enhancing communication, and to advancing the state of the art of FOSS4G. The importance of this role cannot be emphasized enough. Applications and studies that are not properly documented and analyzed have little or no impact on highlighting current challenges and needs for improvement, pointing to promising directions for future developments, disseminate the benefits and challenges of using FOSS4G, and promoting the use of FOSS4G in suitable applications and contexts. The FOSS4G community must keep this need in the forefront of its priorities and avoid the mistake of developing outstanding software and applications that are "best kept secrets" to most of the end users, developers, businesses, governments, educators, and researchers that can benefit from their features and development philosophy.



FOSS and FOSS4G under certain circumstances can be superior alternatives to their proprietary counterparts (Moreno-Sanchez et al. 2007, Wheeler 2007). The reasons for FOSS adoption should be based not only on their no-cost feature or the access to the source code. FOSS and FOSS4G should be evaluated on par with commercial off-the-shelf (COTS) software following the criteria suggested by Wang and Wang (2001) and Ven et al. (2008). A growing number of commercial services, studies, and resources are available to assist potential users in choosing and deploying the best FOSS/FOSS4G for their specific informational needs (e.g. Holck et al. 2005, Woods and Guliani 2005, Ven et al. 2008, The FOSS Evaluation Center², OpenGeo³). Though findings are varied as to the strengths and weaknesses of FOSS for specific contexts and purposes (Erlich and Aviv 2007, Ven et al. 2008), today

it is clear that FOSS/FOSS4G not only provide healthy competition for proprietary solutions but also opportunities for mutual benefit and complementarity. There is a need to study and discuss the impacts of these interactions on advancing the current state of geospatial software, as well as on enhancing the delivery of the systems and geospatial information demanded by citizens, businesses, governments, educators and researchers around the world.

The interest in FOSS4G is reflected in the growing number of presentations and increased attendance at the annual FOSS4G international conference. The 2011 event held in Denver had over 900 participants from 42 countries, 24 workshops, and 150 presentations. The presentations and discussions made evident the speed of progress, high level of maturity and advanced capabilities of many FOSS4G projects. This issue is dedicated to papers selected from the Academic Track of this conference. These papers were reviewed and chosen for this issue by the 2011 International Scientific Committee of this track (See closing section Appendix 1).

In the first paper Butt and Li present a system that incorporates FOSS groupware and the delivery of maps in real time over the Web for enhancing public involvement in decision making in Canada.

In the second paper, Cavner et al. from Kansas in the USA describe their plans and ongoing development efforts for producing macroecology and biogeography tools dealing with large species presence data structures using the Open Geospatial Consortium Web Processing Service (WPS) specification and Quantum GIS (QGIS) as a WPS client.

In the third paper, Hollander describes how a non-hierarchical clustering algorithm (Partitioning around medoids or PAM) is used in California USA for landscape regionalization making use of GRASS⁴ and R⁵.

Next, Donchyts et al. from the Netherlands developed a new Application Programming Interface (API) which simplifies working with geospatial coverages as well as other data structures of multi-dimensional nature. The API is made available as a set of Open Source code libraries in C#.

In the fifth paper, Jennings describes how difficult economic times in California USA forced a community college to explore options and choose FOSS4G Remote Sensing (RS) software, Optiks⁶, to teach their RS course successfully without reducing the breadth and depth of the knowledge and skills imparted in the course.

In the next paper Boerboom from the Netherlands presents the challenges and future directions for integrating web-based services from the GIS and decision sciences domains. He uses as case study the development of forest management plans for adaptation to climate change in Europe. The formulation of these plans takes place in multiple regions and requires multi-criteria evaluation of vulnerability and adaptive options.

Finally, Knudsen et al. from the Danish National Survey and Cadastre describe the geographic transformation system

²The FOSS Evaluation Center <http://foss.technologyevaluation.com/>

³OpenGeo: <http://opengeo.org/products/suite/>

⁴GRASS GIS: <http://grass.osgeo.org/>

⁵R Project: <http://www.r-project.org/>

⁶Optiks: <http://opticks.org/confluence/display/opticks/Welcme+To+Opticks>

used in this agency to transform Danish map projections and datums, and how it has been recently integrated into PROJ⁷ the leading Open Source cartographic projections library.

The papers in this issue are part of the exponential growth in the number of studies and publications documenting applications of FOSS/FOSS4G in diverse areas of expertise and in varied technological, institutional, economic and socio-cultural contexts around the world. The FOSS4G community must continue its efforts to document and give wide dissemination to these experiences and developments to advance the state of the art of FOSS4G and to help it attain the recognition and place it deserves in the Information Technology and geospatial technologies world. The existence of the OSGeo Journal plays a major role in achieving these goals.

References

- CRM-Reviews 2006.** 50 Open source success stories in business, education and government CRM Articles October 31, 2006 [WWW Document](#)⁸
- Erlich, Z. and Aviv, R. 2007.** Open Source Software: Strengths and weaknesses. In St. Amant K and Still B (eds) Handbook of research Open Source Software: Technological, economic and social perspectives. Hersey PA, IGI Global pp. 184-196.
- Faber, S. 2007.** Geoserver and Open Standards: A success story FOSS4G 2007 September 24-27, 2007 Victoria, Canada (available online at 2007.foss4g.org)⁹
- Garbin, D. and Fisher, J. L. 2010.** Open Source for enterprise Geographic Information Systems. IT Professional November/December 2010: 38-45.
- Holck, J., Persen M.K., and Larsen, M.H. 2005.** Open Source Software acquisition: Beyond the business case. In 13th European Conference on Information Systems 26-28 May 2005 Regensburg, Germany. (Available at [Open Source Software acquisition](#)¹⁰)
- Moreno-Sanchez, R., Anderson, J., Cruz, J. and Hayden, M. 2007.** The potential for use of Open Source Software and Open Specifications in creating web-based cross-border health spatial information systems International Journal of Geographical Information Science 21: 1135-1163.
- Ven, K., Verelst, J. and Mannaert, H. 2008.** Should you adopt Open Source Software IEEE Software May/June 2008: 54-59 (Available at [Should you adopt Open Source...](#))¹¹
- Wang, H. and Wang, C. 2001.** Open Source Software Adoption: A Status Report. IEEE Software 18: 90-95.
- Wheeler, D.A. 2007.** Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers! [WWW document](#)¹²
- Weber, S. 2004.** The success of open source Cambridge MA, Harvard University Press.
- Woods, D. and Guliani, G. 2005.** Open Source for the Enterprise Managing Risks, Reaping Rewards. Sebastopol, CA, O'Reilly & Associates.

Appendix 1: Members of the 2011 FOSS4G Conference Academic Track (in no particular order)

- Tyler Erickson** Michigan Tech Research Institute, USA
Serena Coetzee University of Pretoria, South Africa
Maria Brovelli Politecnico di Milano campus Como, Italy
Helena Mitsova North Carolina State University, USA
Thierry Badard Laval University, Canada
Venkatesh Raghavan Osaka City University, Japan
Songnian Li Ryerson University, Canada
Jeff McKenna Getaway Geomatics, Canada
Rafael Moreno University of Colorado Denver, USA

⁷PROJ: <http://trac.osgeo.org/proj/>

⁸CRM-Reviews 2006: <http://www.crm-reviews.com/50-open-source-success-stories-in-business-education-and-government>

⁹Geoserver and Open Standards: http://2007.foss4g.org/presentations/view.php?abstract_id=8

¹⁰Open Source Software acquisition: <http://csrc.lse.ac.uk/asp/aspecis/20050130.pdf>

¹¹Should you adopt Open Source...: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4497765>

¹²Why Open Source...: http://www.dwheeler.com/oss_fs_why.html

Open source based online map sharing to support real-time collaboration

Muhammad A Butt, Songnian Li

Abstract

Collaboration is an important part of many tasks involving people from different organizations, in which maps often play a central role in informing and improving debates and facilitating decision making. Allowing groups to share and view maps and spatial images interactively over the Web in real-time not only provides an effective solution to decision makers, but also facilitates scientific and public debates with real-time geospatial information. A few tools have been developed using proprietary software approaches, e.g., PCI Geoconference. More recently, some efforts have been made using open map services to develop simple map sharing applications. However, little has been done on designing and developing such online tools based on open source. Further, a literature search indicates the lack of scientific publications on empirical studies of their practical applications. This paper describes a study on using Open Source Geographical Information System (OSGIS) and mapping solutions to design and develop real-time map sharing applications, which rely on the data served through open map/data services with the option of integrating local data. The study focuses on not only real-time map (or geospatial information) sharing, but also the integration of other open source based groupware solutions. Existing open source solutions are evaluated for the design and development of various prototype collaborative map sharing tools. The prototype is applied in an online virtual public meeting space for initial usability studies. The paper also discusses the issues related to the design, data required to support better map sharing, and adoption of related standards.

Introduction

Multi-user collaboration is increasingly integrated in many tasks involving people from different organizations, in which maps often play a central role for providing visual information to support collaborative decision making. The rapidly expanding range of Web technology has made it possible to collaboratively make decisions over the Web. Demands for web-based open mapping Application Programming Interface (API), integrated with other information and Computer Supported Cooperative Work (CSCW) tools, have rapidly become more important for supporting real-time map sharing solutions. Therefore, the establishment of real-time collaborative map-based applications is one positive step taken by the researchers that are progressively working in many fields, for example, urban planning projects, emergency system, GIS data production, etc., that incorporate collaborative involvement (Chang 2010; Al-Kodmany 2002; Huang 2001; Roseman 1992; Brail et al. 2001; Klosterman 2001).

A synchronous approach is developed to support collaboration among users (Chang 2010); however, little has been done on designing and developing such Open Source Soft-

ware (OSS) - based online map sharing tools to support real-time collaboration. By examining the researchers' contributions from the literature review, this study seeks to outline the significance of ensuring the implementation of valuable and adequate methods, techniques and tools to fill the research gap. Multi-user synchronous communications and/or discussions among the participants and between the participants and decision makers often improves the understanding that leads to effective feedback and enhanced decision-making (Evans et al. 1999; Ventura et al. 2002; Tang 2006; Bryant et al. 2006; MacEachren et al. 2001; Li et al. 2007; Jankowski and Nyerges 2001; Jankowski and Nyerges 2003). This study expects to develop an open mapping API-based real-time collaborative infrastructure with the option of integrating local data for enhancing involvement during debate. The study focuses on not only real-time map (or geospatial information) sharing, but also integration of other open source based groupware solutions. The objective is to make sure that the model with synchronous collaborative support of information and map sharing mechanisms will help to improve/increase participants' involvement and/or aid decision makers in reaching a final decision efficiently (Chung et al. 1994; Begole et al. 1999).

The study also models and presents the prototype development of an integrated online synchronous collaborative system by putting together the practical integration of various OSGIS, Web GIS, OSS-based tools and open mapping APIs. Some of this research prototype's components are still in development and in early stage of testing. The core purpose of developing such a model is to allow interested groups to share and view maps and spatial images interactively over the Web in real-time: for instance, providing adequate access to real-time collaboration tools (i.e., real-time map sharing for exploring spatial context) to provide information and data as maps and visualizations, which the users can explore in order to make better choices (Li et al. 2007; Li et al. 2006; Li et al. 2004; Laurini 2004; Dragicevic 2004; Laurini 1998; Obermeyer 1998; Rinner 1999; Edelenbos 1999). This study explains certain facts or observations (i.e., core concepts, design and testing, etc.) with an overview of enabling technologies for analyzing and designing a successful mechanism. Moreover, it describes a prototype development based on a research project that looks into integrating CSCW principles and open source groupware tools with web-based GIS.

The enhanced OSS-based prototype integrates social collaboration tools, Web-mapping functionalities, and Human Computer Interaction (HCI) principles. In brief, the prototype system is the integration of various types of open source based modules and open map services i.e., Google, Yahoo, etc. to provide synchronous-based Collaborative Real-time Map sharing Infrastructure (CRMI) to support and/or enhance input during planning and decision making related workflows.

In addition, as a part of the study, there is a plan of testing the prototype's usability by providing a case study and/or pilot project involving participants concerned with planning

issues at the York Regional municipality in Canada. The municipality data are provided and acquired by the municipality Website (<http://www.york.ca>) to create a mock case study for in-house usability testing (see Section 4.1 for more information about the study area and selected sites). In authors' opinion, Web-based synchronous GIS applications need consistent usability evaluation during development and after development regarding end-user's requirements. Therefore, if authors get any chance of an approval and/or agreement from municipal bodies (of York) for conducting the test in a real scenario at later stages (of this study), then the prototype usability testing will be conducted again to get more precise results. The main purpose of this usability evaluation is to find out the results which will depict/conclude that the synchronous collaborative map sharing framework, when integrated and developed with Internet-based GIS and OSS technologies, can provide cost effective solutions and significant support in enhancing real-time participation as well as improving decision making process.

It is hypothesized that with the lower cost and more effective real-time collaborative communication channels, partially due to the adoption of open source solutions, the prototype would help increase the degree to which the citizen, local bodies, environmental assessment, government agencies and decision authorities work at the same time.

Background-Related Work

The CSCW application or groupware technologies allow people in remote places to interact with each other by sharing the documents and files through voice, data and video links (Scientific 1990; Antunes et al. 2009; Abdalla et al. 2010). Baecker (1993) defined groupware as information technology used to help people work together more effectively. The decision making process is based on group and collaborative activities, however, the procedures that used in GIS have been developed for use by individuals. Armstrong (1994) states GIS applications are not well designed for collaborative activities (e.g., decision making). A few GIS-based tools having groupware and CSCW functionalities have been developed using proprietary software approaches, e.g., PCI geo-conference (see Section 3.3 for detailed comparison between authors' prototype and PCI Geo-conference – a commercial tool). More recently, some efforts have been made using open map services to develop simple map sharing applications.

Rinner (1999), Rinner (2006) and Li et al. (2007) realize the need to support collaborative discussions by introducing asynchronous-based geo-referenced mapping frameworks in which the discussion thread of each individual is linked with one or many elements on the map. These techniques offer a limited way of exploring spatial data or map information collaboratively, while current study is focusing more on providing ways in which participants solve spatial problems together using synchronous communication.

Several prototypes that facilitate anywhere (real-time) collaboration are designed and developed as a result of recent advancements in Geographic Information Technology (GIT) that support large spatial databases, groupware technologies and Web-based GIS (Churcher 1999; Jones et al. 1997; Dragicevic et al. 2004; Boulos et al. 2010). For instance, Spatial Group Choice, a spatial decision support prototype

was developed by Jankowski et al. (1997) to support the CSCW technique. Similarly, GroupARC was proposed and developed by Churcher (1996) which provides a tool to geographically scattered people to collaboratively view and annotate map/spatial data. The prototype Real-time Environment Information Network and Analysis System (REINAS) was designed and developed by Pang (1995) which includes functionalities that are useful in the analysis of geospatial data. Virtual Emergency Operations Center prototype was developed which aims to provide a collaborative virtual environment that enables interactivity among participants while executing synchronous, script-driven tests and simulations (Fiedrich et al. 2007). Chang (2010) developed Synchronous collaborative 3D GIS to support synchronous collaboration efforts among geographically-distributed people for enhancing collaborative decision-making.

Research related to the design and implementation of real-time collaborative mapping technologies is still at an early stage of development. There is not much literature in this field. As a result, there have been only a few and/or inadequate empirical studies addressing the synchronous collaboration, real-time map sharing mechanism and procedure of group spatial decision making designed to facilitate collaborative work (MacEachren 2000; Boroushaki et al. 2010). The rapid changes in technology, especially, in the field of GIS, OS-GIS, GIT, CSCW, groupware, Internet computing technology, virtual reality, geographic visualization, geo-collaborative technology and others will have a significant influence on the shift and/or merge of those technologies into collaborative synchronous GIS.

Design and Development of OS-based Collaborative Map sharing Tool

The CRMI aims for a combined action or synergy of multiple components that are responsible for a variety of unit functions using two tiers of communication protocols. The presentation tier is called the CRMI protocol suite which runs over HTTP/S and is designed for real-time communication between users (e.g., decision makers) with the help of Web console. The application tier which constitutes CRMI internal architecture is based on OSS-based components, modules and services.

CRMI Architecture

The CRMI architecture is mainly built by using open source modules and services. Figure 1 depicts the CRMI architecture.

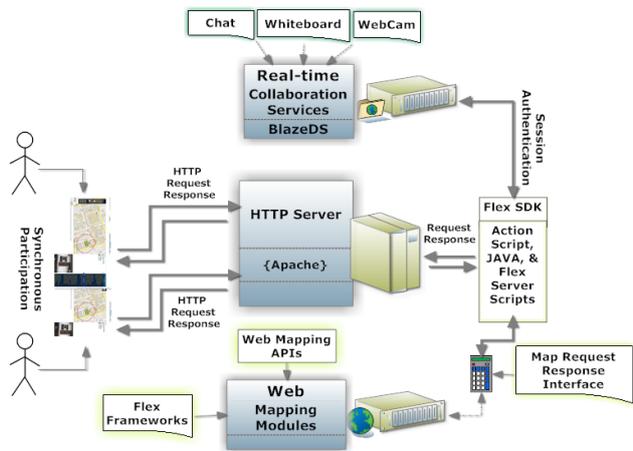


Figure 1: Architecture of collaborative real-time mapping infrastructure

The CRMI backend architecture is internally comprised of three major components that are linked with each other for two-way communications by using Adobe flex framework and JAVA scripting modules. These backend components along with the HTTP server component collectively form the Collaborative Real-time Map Sharing Infrastructure, which are briefly discussed below.

Real-time Collaboration Services: provide the infrastructure which enables developers to add real-time collaboration capabilities (real-time push messaging) to their applications. Collaboration service is JavaScript Object Notation over HTTP/S based system that leverages Web 2.0 messaging and interaction paradigm (Lemos 2006). It is usually commoditized by using Red5 open source media server that uses real-time multimedia protocol to support audio/video streaming (Wang et al. 2010).

Blaze Data Service (BlazeDS): provides real-time Web messaging capabilities (a complete publish/subscribe infrastructure allowing Flex clients and the server to exchange messages) for rich Internet applications using ActionScript message format over HTTP/S.

Web Mapping Modules: handle the user’s request on the maps displayed on users’ screen by producing open map APIs (Google, Yahoo, and Bing). The synchronous behavior of maps is created to enable co-browsing (co-shared maps) among the participants’ Web console (user’s screen) by using Adobe Flex framework, map API for flash and Action/Java script programming language. In addition, the open source Flex software development kit provides an integration link between real-time collaboration services and map APIs modules for the development and deployment of CRMI.

HTTP Server: handles the user’s requests sent from front-end Web interface (i.e., Firefox or Internet Explorer) to the Web server (Apache) and then response back (i.e., an interface along with real-time map sharing functionalities) to the Web client using HTTP protocols.

OSS-based Collaborative Map Sharing Tool

A prototype as a proof of the concept was designed and developed to support the collaboration in multi-user environment for informing and improving debates and facilitating decision making. Two major components of the prototype which sup-

port real-time map sharing mechanism are briefly discussed as follows:

1) Collaborative Map Sharing Component

A collaborative map sharing component was implemented with some basic mapping functions to explore planning and development scenarios within geographical contexts of the projects. A picture is worth a thousand words. For many planning and decision oriented discussions, having a map or map-based displays of different scenarios shared by participants greatly facilitates their discussion on some issues. For that reason, a collaborative map sharing component was developed which allow participants and decision makers to collaboratively explore geographic contexts of the projects while discussing issues. Figure 2 shows the main Web interface of the collaborative map sharing component.

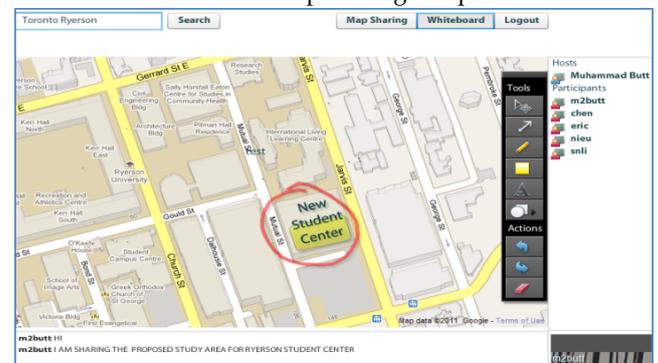


Figure 2: Real-time collaboration using Google mapping API

The component integrates a number of features, including map sharing, chatting, video conferencing and whiteboarding, into a single interface that can be invoked and run in a new window. It is intended to be used by a group of participants who have a common topic of discussion, which requires access to map displays to make their points clearer. Any participant can initiate a collaboration session and invite others to join. All participants who join the session are given the right to use white-boarding tools for creating drawings as well as annotating on a Google map. For example, a free hand drawing object (new student center building area) on the map can be created by one participant and shared by all participants during the meeting session (see Figure 2).

This synchronous (real-time) collaboration function is currently implemented using Google maps as the base map, and authors are working on how to add project-related GIS data into the display. Another feature is to record the session history so it can be replayed at a later time, a feature very useful to later comers.

2) Collaborative Geo-conferencing Component

The virtual geo-conferencing with real-time map sharing is a supplement to the real meetings to give those who cannot physically attend the conference and/or seminars a chance to participate (Boulos et al. 2010; Haklay et al. 2008; Cammack 2007; Doyle et al. 1998). The idea is to stream real meetings online and provide facilitating tools to enable online users to question and interact, and to allow presenters to integrate their electronic presentations and maps (Google, Yahoo, etc.) into the virtual conferencing environment. Figure 3 shows a view of the virtual collaborative geo-conferencing interface.

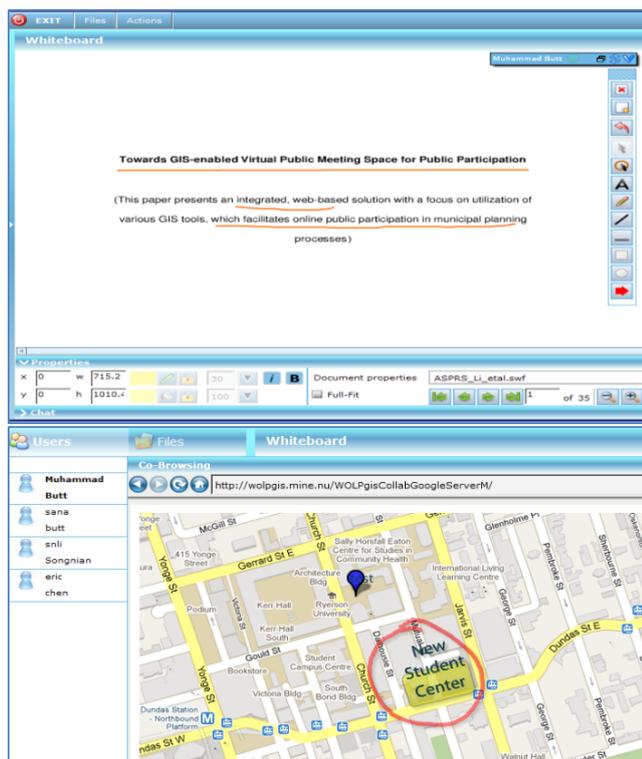


Figure 3: Virtual collaborative geo-conferencing interface with real meeting streams

By incorporating the collaborative map sharing components, the interface allows the presenters to share map displays in the same way as PowerPoint slides are shared on screens in many other web conferencing/groupware systems. The presenters can also use the built-in whiteboard tools to select features, add annotations, and draw graphics to improve debates among multi-users. However, during the virtual conferencing, only the presenters can initiate the tools and control how the map data should be displayed to facilitate their presentations.

Comparison between PCI Geo-Conference and CRMI Geo-Conferencing Component

Table 1 presents a comparison chart constructed using the PCI Geomatics geo-conference application and the developed CRMI geo-conferencing components' features. Both systems were designed to provide an innovative way to share maps, data and images with geographically spread participants in real-time that enhance the participatory approach in the planning and development related discussions. Intended use of these systems is to facilitate the GIS professionals, the decision authorities, the non-professionals and the public to increase participation and enhance decision making. Some of the features of the PCI geo-conference application are unknown, for instance, its support to open map APIs, i.e., Google, Yahoo, Bing, MapQuest for maps sharing as well as its support to documents such as PDF, Microsoft PowerPoint and Word for real-time file sharing. Another major difference is that CRMI geo-conferencing component is purely Web-based (needs Web browser to initiate the session), whereas PCI geo-conference

is built around two main components: 1) the geo-conference client application, and 2) geo-conference server. The users access and participate in Internet mapping conference sessions via the geo-conference client which needs to be installed and configured (on the user's machine) in order to connect or participate during the participatory session.

Usability Study of the Prototype Components

Usability Evaluation of CRMI: A Plan of Implementing A Mock Case Study in Regional Municipality, Canada

Study Area: To validate or test the research prototype's usability, an existing urban expansion (future amendment) project in York Region, Canada, was selected as a case study. York regional municipality covers 1,776 square kilometres from Lake Simcoe in the north to Steeles Avenue in the south. It borders Simcoe county and the Region of Peel in the west and the Durham Region in the east. York Region has nine municipalities: Georgina, East Gwillimbury, Newmarket, Aurora, Whitchurch-Stouffville, Markham, Vaughan, Richmond Hill and King. The region is very diverse with quaint rural villages, bustling suburban communities and vibrant cosmopolitan urban areas. With a population over 1 million and with one of the nation's highest growth rates, the region is a rapidly growing and changing area. As a result of growing population in the region, issues related to urban planning have become increasingly important.

Selecting Sites in York Region: Based on the by-laws and like other municipalities and regions in Canada, York is required to hold meetings regarding forthcoming projects related tasks undertaken by York Regional council. Notices are posted on the region's website, as well as advertised in local municipality newspapers. The planning department of regional municipality of York has selected three potential sites for amendments in the town of East Gwillimbury, Markham and city of Vaughn in York Region. Notice of the first prehearing conference appeared in newspapers with general circulation in York Region on March 24, 2011. The planning department welcomes any participation with suggestions or recommendations from local residents and/or citizens regarding potential/pre-selected locations for future urban expansion or development.

Amendment 1 to the York Region Official Plan - Urban Expansion in the Town of East Gwillimbury: This amendment proposes to expand the urban area of the Town of East Gwillimbury to provide opportunities for urban growth to the year 2031. The lands subject to this amendment (was adopted by regional council on September 23, 2010) are shown in the key map (see Figure 4).¹³

¹³Source: <http://www.york.ca/Departments/Planningand+Development>

Category\Product	CRMI	GeoConference
Map sharing		
Support Open Map APIs	☒ Support Google & Yahoo Maps	○ - unknown
Support MapServer/WMS/IMS	●	●
Open Source		
Require License to deploy	● - Open Source	○ Require License
Available Open Source Code	●	○
Web-based		
100% Browser-based	●	☒ - Need Client to be installed
CMS Integration	●	○ - unknown
Data sharing(PDF, PowerPoints)	●	○ - unknown
Screen Sharing	●	○ - unknown
Feedback		
Send News Letter	●	○ - unknown
Send Meeting Invitation	●	●
Review Capabilities		
Record Session	●	●
Audio/Video Conferencing		
Audio/Video Chat	●	○ - unknown
Meeting Manager		
Administration Meeting Room	●	●
Management of Users	●	●

Legends ● Supported ☒ Partially Supported ○ Not Supported/Unknown

Table 1: Comparison chart for PCI geo-conference and RCMI geo-conferencing component

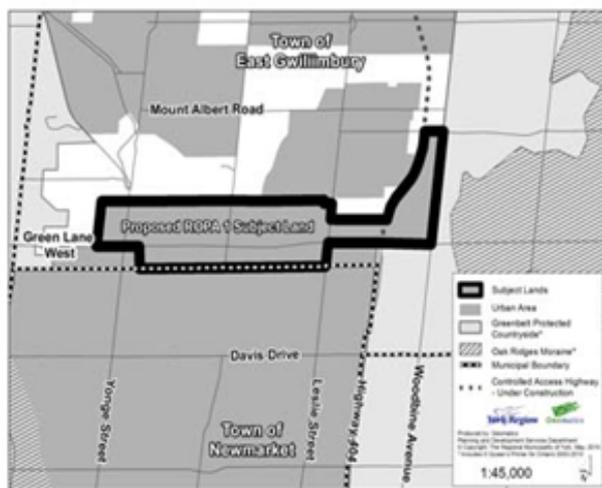


Figure 4: Urban Expansion in the Town of East Gwillimbury

Amendment 2 to the York Region Official Plan - Urban Expansion in the City of Vaughan: This amendment proposes to expand the urban area of the City of Vaughan to provide opportunities for urban growth to the year 2031. The lands subject to this amendment are shown in the key map (see Figure 5).¹³ This amendment was adopted by regional council on September 23, 2010 and was subsequently appealed to the Ontario municipal board.

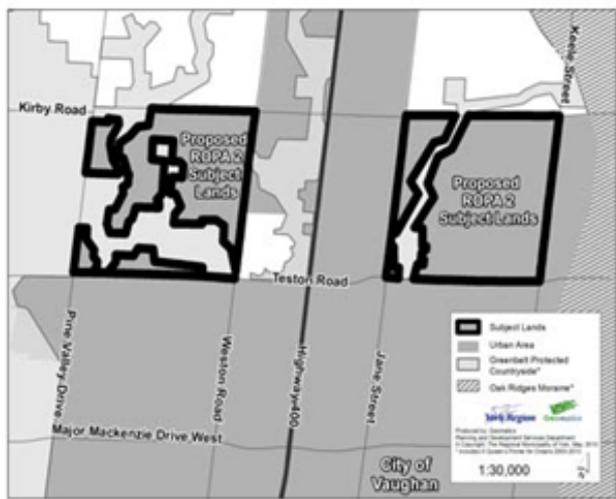


Figure 5: Urban Expansion in the City of Vaughan

Amendment 3 to the York Region Official Plan - Urban Expansion in the Town of Markham: This amendment proposes to expand the urban area of the Town of Markham to provide opportunities for urban growth to the year 2031. This amendment was adopted by regional council on September 23, 2010 and shown in the Figure 6.¹³

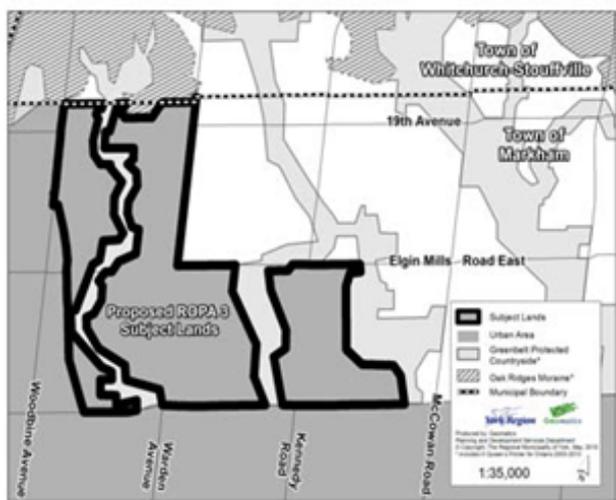


Figure 6: Urban Expansion in the Town of Markham

Evaluation of CRMI

The described context of the case study is used to evaluate and test a system prototype. The system is targeted for different participants or attendees involved during the planning and development related workflows to enhance public involvement. The following explains further the procedures/steps taken during the usability evaluation of the research prototype components as there is a prospective plan of implementing a mock case study (at above mentioned sites) using the urban expansion projects from the Region of York to validate the proposed approach.

Development of Evaluation Criteria

This section discusses the aspects, based on which the prototype is to be evaluated. These aspects mainly concentrates on: 1) intended users; 2) initial cost; 3) interactivity; 4) effective participation; 5) real-time communication; and 5) usefulness. The criteria for the evaluation of prototype implementation are somewhat similar to the ones adopted by different researchers, such as Chang (2010), Tang (2006), Zhao and Coleman (2006), Rinner (2006) and Ma (2006). Answering the questions in connection with the evaluation criteria can also indicate to what extent the (level of) successful implementation of the developed prototype is possible.

- 1) **Intended Users:** Does the prototype provide a virtual platform to facilitate the non-professionals and citizens for participation during the planning related discussions?
- 2) **Initial Cost:** The prototype was developed using open source software, open source components/modules and open source GIS technologies, therefore, the initial cost of the prototype development is almost absent. Does the prototype provide a cost effective solutions to smaller municipality regions?
- 3) **Interactivity/HCI:** Does the prototype provide user-friendly and interactive user-interfaces? How quickly can participants communicate with the consultants/mediators and receive a response back for their queries?
- 4) **Effective Participation:** As compared with the existing applications, Can the prototype fulfill participation (public involvement) needs in a more efficient manner? To what extent? Can the prototype fulfill participation needs for spatially related discussion during planning? Is the multi-way of participation possible among users as well as higher authorities? Do the proposed participation approaches automate the planning workflows and/or improve decision making significantly? To what extent?
- 5) **Real-time Collaboration/Communication:** To what extent are Web GIS-based Virtual Public Meeting Space collaboration components (i.e., collaborative map sharing and geo-conferencing) beneficial in facilitating the moderator to share his/her ideas, documents, mapping information, spatial data and decisions related to development among other participants or meeting attendees on a real-time basis?
- 6) **Usefulness/Connectivity:** Does the system allow participants to easily access additional information and documentation necessary to provide input to the project under consultation? Could this platform be useful for York Region’s future amendments? What could be missed in prototype functional aspects in terms of user-friendliness? What could be improved in prototype functional aspects?

How to Evaluate

The evaluation is conducted in three parts: 1) a pre-questionnaire with queries about the user’s background, computers expertise and GIS knowledge; 2) using analytical method with the help of Google Analytics tool; and 3) final questionnaire/CRMI polling component for collecting feedback concerning the usability of the system. The procedure is described in detail as follows:

1. The pre-questionnaire will contain some questions about the user’s experience of other Web GIS applications, for in-

stance, questions about their GIS experience, finding store location/address, Web surfing experience and public participatory planning experience besides the questions about the user's background.

2. As mentioned above, the second part of the evaluation is the actual interaction with the system. In addition, to evaluate the usability of the prototype and assess the degree of public involvement in the online virtual public meeting session, meeting attendees' interaction and each move on the prototype interfaces will be recorded using Google Analytics. Google Analytics is an easy to use, free, and very handy usability testing tool that provides a comprehensive set of website data tracking and analysis tools. Usability researchers who want to understand user/participant behaviors can use these tools to gather a great amount of usage statistics and reports (Atterer et al. 2007; Aditya 2010). By using the Google Analytics, it is possible to collect highly detailed and useful data about the actual usage of the prototype and its components. Data includes user sessions, visits, page views, button clicks, audio and video capturing, and the user's screen (through a desktop-streaming tool) are automatically captured and/or recorded. These data elements are useful for evaluating the usability as well as measuring the degree of public input during the process of participatory planning and decision-making.
3. Finally, the users are asked to fill out a questionnaire at the end of the online participation virtual meeting session. Questions related to ease of use or interactivity of using Prototype interfaces will be included in this questionnaire.

Banati et al. (2006) states two approaches of prototype evaluation: (1) an inspection method usually performed by system developer, and (2) a user testing when actual users are involved to walk through the system's functionality. Therefore, the evaluation criteria 'outcomes' may not be measured precisely, until the system is deployed and evaluated with a real scenario/case related to any project (Bevan 1995; Lantz 1986; Carl et al. 2005) or by addressing the above questions during the system's usability evaluation that follows any IT-based testing standards (i.e., ISO or IEEE, etc.). In brief, the study goal, developing a GIS-based synchronous map sharing mechanism to involve the user during discussion, will only be achieved if the answers to the majority of the above questions (discussed in evaluation criteria) are favorable.

Evaluation results of this research are not presented in this paper as prototype's usability testing has not been fully accomplished up to this point. No doubt, the ongoing usability testing will more be focused on measuring the prototype's capacity to meet its intended purpose.

Related Discussion

This study introduces collaborative methods to encourage a synchronous approach that support users' involvement during planning and decision-making processes. Although some components of a prototype are designed and developed as a proof of the concept, which support real-time map sharing mechanism along with functionalities of groupware tools. There are still some key design and implementation issues, which need to be further studied and resolved in the next-step of the ongoing research, before its deployment into the real-world scenario testing (i.e., system usability and performance

evaluation, etc.). Some of the implementation issues faced during the study and preliminary prototype development is discussed in the following paragraphs.

The authors also expect some sociological issues and forthcoming risks, which may occur after the system deployment, e.g., facing difficulties by the users (public, city staff, project proponent, etc.) for accepting and responding to the Web GIS, multimedia and groupware technologies in connection to traditional practice. These potential hindrances also led authors to use the modular/component-based design implementation (allow and support the incremental implementation of the developed methods), which may be helpful in acquainting users with the shift of new technologies. However, this issue might need further inspection at later stages of the study.

One of the organizational issues is to deal with the privacy of data, which was another concern during the design and development of the prototype. It necessitates the needs for protocols to be implemented that manage the privacy of project data in connection to its copyright protection law. Setting the rights and privileges of the users at the different levels of their interaction with the system may resolve this issue.

Another big issue is to be considered during the system development in terms of interface designs, interactivity, and quick system responsiveness. In addition, special consideration is taken while providing interactivity and user-friendly interfaces by following a common set of rules of HCI. We anticipate that with the implementation of HCI principles, such as strive for consistency using cascading style sheet for fonts and colors, informative feedback using Java-based dynamic effects of tooltips and error prevention using JavaScript-based popup messages, the system can provide more efficient, interactive and user-friendly interfaces to the end-users. The prototype is composed of different participation components, which require or involve some sort of GIS-based functional integration (some technical knowledge is required to better utilize the tools). Therefore, design of such interfaces by keeping in mind the expertise of the naive users or non-technical participant becomes a big challenge during development.

The prototype developed is an effort of technology integration from a technological perspective. Some (main) key design issues related to technologies used during the development of the prototype are described as follows:

As the mapping component was designed initially for the proof of the concept, GIS mapping functionality over the base layer (Google map) is limited, which needs to be enhanced by embracing more carefully-selected functions. Moreover, the real-time (synchronous) collaborative participation component is currently developed and implemented using Google maps as the base map, whereas, it should be able to use any open map APIs (Yahoo, Bing, MapQuest, OpenStreet, etc.) to implement synchronous participatory behaviors and/or function based on the presented architecture for sharing spatial data during discussion and/or debate. As discussed earlier, in the future authors are working on how to add project-related GIS data as a background layer into the display.

The collaborative real-time map sharing component was initially designed for enabling "spatial" virtual conferencing along with WebGIS support. This component was developed using Google map API, adobe flex, and flash collaboration service technologies. The Adobe Flex platform provides and builds real-time collaboration-enabled applications rapidly

as it offers a complete software development kit, which contains ready-to-use components, designed with collaborative workflow. The issue at this moment is that flash collaboration services in the Adobe platform does not support screen or media sharing protocols (another design challenge, which is not considered during the initial development). The purpose of this function is to record the session history during the conference/seminars so it can be replayed and shared at a later time. However, the issue might need further inspection at later stages of the study.

The collaborative geo-conferencing component (having combined functionality of collaborative map and documents media sharing) was designed and developed using JAVA and Adobe Flex platforms. Integration between two platforms was a challenge in relation to their suitability, connectivity and scalability in the beginning of the development. For instance, Flex code requires a flash player to run or execute, whereas Java requires Java Virtual Machine (JVM) to execute the compiled code. Other problems common to all online application developments are the compatibility of different web browsers (i.e., Internet Explorer, Mozilla and Google Chrome, etc.) and Internet connection speed. The benefit of using a JAVA platform will resolve the issue related to screen and video sharing protocol of Adobe flash collaboration services. Therefore, the share/record screen module will be developed and added as a recording function along with a meeting planner component (useful to the city staff for scheduling public meeting reminders or auto-notification) using JAVA as a part of the virtual public meeting interface at later stages of the study and prototype development.

Finally, the usability evaluation and performance testing for measuring the functionality of the prototype during the case study (at York Region) will be an important future task of this research, which needs to be performed for evaluating important aspects: (1) to determine whether or not the designed framework will help in improving the public participation during the planning process by maximizing public substantial input; (2) to evaluate that to what extent the system will be usable and to fulfill participatory needs of the current practice; (3) to deduce how the public responds in adopting the GIT-based means of participation; (4) to assess the amount of training required for non-technical staff and/or participants to use the system's components efficiently; (5) to find out how the city staff, higher authorities, and project proponent handle the citizen's feedback in timely fashion for the quick and/or effective decision-making and determine how quickly the system responds to participant; and (6) to find out how citizens' access to information, communication channels, level of public participation, and the overall decision-making process may be impacted or influenced by adopting the prototype, in relation with traditional participatory approaches (Li et al. 2007). In addition, to get the proper feedback and have the answers to all above-mentioned questions it is necessary to implement the system in a real world scenario as a case study and/or pilot project for part of the study.

Conclusions

The study reported in this paper brings together OSGIS, groupware, other web-based information technologies, and open source mapping APIs-based solutions to design and de-

velop real-time collaborative map sharing components, which rely on the data served through open map/data services with the option of integrating local data. Armstrong (1994), Craig (1998), Craig (2002) and Baker et al. (2005) state more collaboration among users involved during developing the plan, the more likely that the plan will appropriately address issues that are important to the bodies. The prototype aims at providing a web-based virtual conferencing environment that encourages multi-users to get involved.

CRMI map sharing and geo-conferencing components have been presented as a concept for enhancing public involvement and aids decision support in spatial planning related tasks. The effectiveness of such a system in supporting real-time collaboration, especially using a synchronous participatory approach that provides a virtual conferencing platform through which live meetings/seminars, spatial data and information can be accessed anywhere and anytime on a real-time basis. Furthermore, it gives users online access to their work, convenient Web-based mapping, and the ability to increase collaborative decision-making via real-time participatory functions, i.e., open API-based map sharing, screen sharing, seminar recording, video streaming, and easy project documents sharing (Bailey 2010).

The prototype framework is designed and implemented using OSS-based technologies to obtain a quality and to minimize a potential cost (i.e., dependence on vendors, a huge early investment and high licensing cost) required to implement the system, which provides the somehow cost effective solutions for the end-users (e.g., small municipalities) with limited or inadequate financial resources. On the contrary, there is a debate among some groups of people about the negative aspects (i.e., stability, scalability, maintenance and reliability, etc.) of the OSS technologies, which was well considered during its selection in the development of the prototype.

During the system development, special attention is given to the selection of enabling OSS technologies as it plays an important role for increasing throughput (quick response), scalability, and design/maintenance cost of the system. For example, the spatial data in the prototype is handled by using PostGIS, the spatial database extension of an open source PostgreSQL Object Relational Database Management System (ORDBMS) that supports broad scale-interoperability with spatial data handling in relation to other open sources ORDBMS (Sano et al. 2003; Wangmutitakul et al. 2003; Wangmutitakul et al. 2004; Wuttiwat et al. 2003). At this stage, due to the lack of availability of the spatial data related to the real projects, the prototype is not demonstrated and configured by using spatial database PostGIS.

The prototype offers user-friendly and self-explanatory functions (with the implementation of HCI rules) for interactive exploration of communication among participants using Web GIS-based virtual meeting participatory platform.

Future work on the prototype will concentrate on the integration of missing features described in the discussion section. The benefits of an integration of OSS-based modules and groupware technologies will be analyzed when the realistic usability testing and evaluation of the prototype (as a proof of concept) is set up or performed, as a mock spatial planning case study for the municipal region, Canada. As the prototype development is based on object-oriented procedures (reusability), further enhancement of the prototype is

practicable.

Acknowledgement

This work reported in this paper was partially supported by the funding from the Canadian National Science and Engineering Research Council (NSERC).

References

1. Abdalla R, Li J (2010) Towards effective application of geospatial technologies for disaster management. *International Journal of Applied Earth Observation and Geoinformation*, 12: 405-407.
2. Aditya T (2010) Usability Issues in Applying Participatory Mapping for Neighborhood Infrastructure Planning. *Transactions in GIS*, 14: 119-147
3. Antunes P, Zurita G, Baloian N (2009) A model for designing geocollaborative artifacts and applications. *Groupware: Design, Implementation, and Use*, 278-294.
4. Armstrong MP (1994) Requirement for the development of GIS-based group decision support systems. *Journal of the American society for information science* 45:667-677
5. Al-Kodmany K (2002) Visualization tools and methods in community planning: From Freehand Sketches to Virtual Reality, *Journal of Planning Literature* 17 (2)
6. Atterer R., Schmidt A. and Wnuk M., 2007, "A Proxy-based Infrastructure for Web Application Sharing and Remote Collaboration on Web Pages", *Proceedings of the 11th International Conference on Human-Computer Interaction*, Rio de Janeiro, Brazil, September 2007, 74-87,
7. <http://murx.medien.fki.lmu.de/~albrecht/pdf/interact2007-usaproxy.pdf> Accessed April 04, 2009
8. Baecker RM (1993) Reading in groupware and computer supported cooperative work: assisting human-human collaboration. San Fransisco, Morgan Kaufman
9. Bailey K, Grossardt T (2010) Toward Structured Public Involvement: Justice, Geography and Collaborative Geospatial/Geovisual Decision Support Systems. *Annals of the Association of American Geographers*, 100: 57-86
10. Banati H, Bedi P, Grover PS (2006) Evaluating Web usability from the user's perspective. *Journal of Computer Science* 2(4): 314-317
11. Baker WH, Addams HL, Davis B (2005) Critical factors for enhancing municipal public hearings. *Public Administration Review* 65 (4):490-499
12. Bevan N (1995) Measuring usability as quality of use. *Software Quality Journal* 4 (2):115-130
13. Brail RK, Klosterman RE (2001) *Planning support systems: Integrating geographic information systems, models, and visualization tools*. ESRI Press, Redlands, California
14. Boulous M, Warren J, Jianya G, Peng Y (2010) Web GIS in practice VIII: HTML5 and the canvas element for interactive online mapping. *International Journal of Health Geographics*, 9: 14-26
15. Boroushaki S, Malczewski J (2010) Measuring consensus for collaborative decision-making: A GIS-based approach. *Computers, Environment and Urban Systems*, 34: 322-332
16. Bryant L, Wilcox D (2006) Some lessons from Web 2.0 for participation and e-democracy. <http://www.headshift.com/archives/002787.cfm>. Accessed June 2006
17. Begole J, Struble CA, Shaffer C, Smith RB (1999) System resource sharing for synchronous collaboration. Technical Report TR-99-11, Computer Science, Virginia Tech. <http://eprints.cs.vt.edu:8000/archive/00000524/01/BetEtAl199b.pdf>. Accessed Oct 2011
18. Cammack R (2007) Cartographic Approaches to Web Mapping Services. *Multimedia Cartography* 1: 441-453
19. Carl C, Warwick I, Neville C (2005) A User Evaluation of Synchronous Collaborative Software Engineering Tools, *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*
20. Chang Z (2010) Synchronous Collaborative 3D GIS with Agent Support. PhD. Thesis, Ryerson University, Toronto, Canada
21. Chung G, Jeffay K, Abdel-Wahab H (1994) Dynamic participation in computer-based conferencing system. *Journal of Computer Communications*, 17(1): 7-16
22. Churcher N, Churcher C (1999) Realtime conferencing in GIS, *Transactions in GIS*, 1999, 3(1):23-30
23. Churcher N, Churcher C (1996) GroupARC – a collaborative approach to GIS, *proceedings of 8th annual colloquium of the spatial information research center, University of Otago, New Zealand, July 9-11, 1996*, pp. 156-163
24. Craig WJ, Elwood SA (1998) How and why community groups use maps and geographic information. *Cartography and Geographic Information Systems* 25 (2):95-104
25. Craig WJ, Harris TM, Weiner D (2002) *Community participation and geographic information systems*. London: Taylor and Francis
26. Dragicevic S, Balram S (2004) A Web GIS collaborative framework to structure and manage distributed planning processes. *Journal of Geographical Systems*, 6: 133-153
27. Dragicevic S (2004) The potential of Web-based GIS. *Journal of Geographical Systems*, 6: 79-81
28. Doyle S, Dodge M, Smith A (1998) The potential of Web-based mapping and virtual reality technologies for modelling urban environments. *Computers, Environment and Urban Systems*, 22:137-155
29. Edelenbos J (1999) Design and management of participatory public policy making. *Public Management Review* 1 (4):569-576
30. Evans A, Kingston R, Carver S, Turton I (1999) Web-based GIS used to enhance public democratic involvement. *Geocomp'99 Conference Proceedings*, Mary Washington College, Virginia, USA, July 27-28 1999
31. Fiedrich F, Burghardt P (2007) Agent-based systems for disaster management. *Commun. ACM* 50: 41-42
32. Haklay M, Singleton A, Parker C (2008) Web mapping 2.0: the Neogeography of the Geoweb. *Geography Compass*, 2:2011-2039.
33. Huang B, Jiang B, Lin H (2001) An integration of GIS, virtual reality and the Internet for visualization, wanalysis and exploration of spatial data. *International Journal of Geographic Information Science* 15(5): 439 - 4
34. Jankowski P, Nyerges T (2001) GIS supported collaborative decision making: Results of an experiment. *Annals of the Association of American Geographers* 91 (1):48-70
35. Jankowski P, Nyerges T (2003) Toward a framework for research on geographic information-supported participatory decision-making. *URISA Journal* 15 (1):9-17
36. Jankowski P, Nyerges T, Smith A, Moore TJ, Horvath E

- (1997) Spatial group choice: a SDSS tool for collaborative spatial decision making, *International Journal of Geographical Information Science*, 11(6):577-602
37. Jones RM, Copas CV, Edmonds EA (1997) GIS support for distributed group-work in regional planning. *International Journal of Geographical Information Science*, 11(1):53-71
 38. Klosterman RE (2001) *Planning support systems*, ESRI Press, Redlands, California, pp 1-23
 39. Lantz K (1986) An Experiment in Integrated Multimedia Conferencing. In: *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '86)*. Austin, Texas: ACM Press. Reprinted in I. Greif (editor), *Computer-Supported Cooperative Work: A Book of Readings*, pp. 533-552, Morgan Kaufmann, 1988
 40. Laurini R (1998) Groupware for urban planning: An introduction. *Computers, Environment and Urban Systems* 22 (4):317-333
 41. Laurini R (2004) Computer systems for public participation. http://www.gisig.it/VPC_sommet/CD_Sommet/ws3/articololaurini.pdf. Accessed 10 January 2011
 42. Lemos M (2006) PHP classes blog: Is PHP ready for Web 2.0? <http://www.phpclasses.org/blog/post/53-Is-PHP-ready-for-Web-20.html>. Accessed 31 May 2009
 43. Li S, Chang Z, Yi R (2004) GIS-based internet noticeboard to facilitate public participation in municipal developments. *Proceedings of the 20th ISPRS Congress, Istanbul, Turkey, July 12-23 2004*
 44. Li S, Guo X, Ma X, Chang Z (2007) Towards GIS-enabled virtual public meeting space for public participation. *Photogrammetric Engineering and Remote Sensing* 73 (6):641
 45. Li S, Ma X (2006) An open source GIS solution for supporting public participation in municipal developments. *Proceedings of the ISPRS Commission IV Symposium on Geospatial Databases for Sustainable Development, Goa, India, September 27-30 2006*
 46. Ma X (2006) A prototype of Web-based PPGIS for municipal planning using open source software. Master Thesis, Ryerson University, Toronto, Canada
 47. MacEachren AM, Brewer I, Steiner E (2001) Geovisualization to mediate collaborative work: Tools to support different-place knowledge construction and decision-making. <http://hero.geog.psu.edu/products/ICC16009amm.pdf>. Accessed 5 June 2010
 48. MacEachren AM (2000) Cartography and GIS: facilitating collaboration, *progress in human geography* 24(1): 445-456
 49. Obermeyer NJ (1998) The evolution of public participation GIS. *Cartography and Geographic Information Systems* 25 (2):65-66
 50. Pang A, Fernandez D (1995) REINAS instrumentation and visualization. *Proceedings, OCEANS '95. MTS/IEEE. Challenges of our changing global environment, San Diego, Oct. 9-15, pp. 1892-1899*
 51. Roseman M, Greenberg S (1992) GroupKit: A groupware toolkit for building real-time conferencing applications. In [CSCW92], (1992) 43-50,
 52. Rinner C (1999) *Argumentation maps – GIS-based discussion support for online planning*. Ph.D. dissertation, University of Bonn, Germany
 53. Rinner C (2006) *Mapping in collaborative spatial decision making*. Collaborative Geographic Information Systems, Idea Group Publishing, Hershey, PA, pp 85-102
 54. Sano J, Wanalertlak N, Maki A, Minoura T (2003) Benefits of web-based GIS/database applications. *Proceedings of 2nd Annual Public Participation GIS Conference, Portland, Oregon, July 2003*
 55. Scientific A (1990) Special issue on communications, *Computers and Networks* 265(3)
 56. Tang M (2006) Design and implementation of a GIS-enabled online discussion forum for participatory planning. M.Sc.E. thesis, Department of Geodesy and Geomatics Engineering Technical Report No. 244, University of New Brunswick, Fredericton, New Brunswick, Canada
 57. Ventura S, Niemann B, Sutphin T, Chenoweth R (2002) GIS-enhanced land-use planning. In *Community Participation and Geographic Information Systems*. London: Taylor and Francis, pp113-124
 58. Wang D, Xu K (2010) Red5 Flash server analysis and video call service implementation, *Web Society (SWS), 2010 IEEE 2nd Symposium on 16-17 Aug. 2010*
 59. Wangmutitakul, Paphun, et al. (2004) WebGD: Framework for Web-based GIS/database applications, *Journal of Object Technology* 3(4): 209-225
 60. Wangmutitakul P, Li L, Minoura T (2003) User participatory Web-based GIS/database application. *Proceedings of Geotech Event Conference, Vancouver, British Columbia, March 16-19 2003*
 61. Wuttiwat T, Minoura T, Steiner J (2003) Using digital orthographic aerial images as user interfaces. *Proc. of ASPRS Annual Conference, Anchorage, Alaska, May 2003*
 62. Zhao J, Coleman DJ (2006) GeoDF: Towards a SDI-based PPGIS application for e-governance. *Proceedings of the GSDI 9 Conference, Santiago, Chile.* <http://www.gsdidocs.org/gsdiconf/GSDI-9/papers/TS9.3paper.pdf>. Accessed 7 Dec 2010
- Muhammad A Butt, Songnian Li*
Dept of Civil Engineering
Ryerson University, Canada
m2butt@ryerson.ca, snli@ryerson.ca

An innovative Web Processing Services based GIS architecture

For global biogeographic analyses of species distributions

Jeffery A. Caver, Aimee M. Stewart, Charles J. Grady, James H. Beach

University of Kansas

jcavner@ku.edu, astewart@ku.edu, cjgrady@ku.edu,
beach@ku.edu

Abstract

Spatial patterns and properties of species richness in natural communities are of keen interest to biogeographers and conservation biologists as they describe key features of the location and distribution of the earth's biological diversity, but species richness tools are scattered across specialty software and are underrepresented in distributed approaches for GIS for work with large datasets. We describe an ongoing development effort for producing macroecology and biogeography tools dealing with large species presence data structures using the Web Processing Service (WPS) specification and Quantum GIS (QGIS) as a WPS client. The creation of species presence/absence matrices is one approach for linking range size and richness patterns and these spatial patterns are well suited for GIS analysis. This paper presents our efforts to date on the development of the Lifemapper Range and Diversity (LmRAD) analysis suite for Lifemapper and on its potential contribution to global biodiversity research and conservation. LmRAD is being engineered as a job based infrastructure that is portable across compute environments for exposing macroecology algorithms for biodiversity calculations as WPS services, a client library for GIS and Scientific Workflow environment software that is tailored for communication using WPS and other OGC standards, and a client plug-in for QGIS. The practical importance of bringing a standardized spatial data processing standard into a distributed GIS environment for macroecology is that larger institutional computer resources can be brought to bear on large problems at vast scales, i.e. continental to global extents at high resolutions for thousands of species. Additionally the decoupled nature of the Web Services approach can allow scientists to mix and match tools in user defined workflows where meta-data can be produced that allow experiment repeatability.

Introduction

The two fundamental units of biogeography are species diversity and the distributional range of species. These two fundamental concepts can be summarized by a basic analytical tool first introduced by Simpson in assessing the diversity of North American mammals (Simpson, 1964). The data for Simpson's study were records of species of recent mammals in quadrants of equal area covering North America,

consisting of a rectangular grid not oriented with respect to physiographic features or other known zoogeographic features. The presence or absence of each species was noted for each quadrant (Simpson, 1964). In this Presence-Absence Matrix (PAM), one axis represents species and the orthogonal axis represents geographic localities or samples. Each geographic site is coded for the presence (1) or absence (0) of each of hundreds or thousands of species resulting in a binary matrix. The PAM combines species richness of sites, the number of species summed across an individual site with range sizes, each species' range expressed as a sum of their presence values across all the sites that they occupy. The PAM has become a basic method used to test ecological and evolutionary hypotheses about the spatial patterns of biological diversity on continental and global scales (Arita et al. 2008), but very few software applications have been developed to aide in PAM construction, analysis and visualization for very large datasets.

We are addressing these challenges by engineering the Lifemapper Range and Diversity (LmRAD) tool. LmRAD is an analysis suite within the current Lifemapper (www.lifemapper.org) platform that will use the computational power of a high-throughput computer cluster to execute macroecology algorithms exposed as Open Geospatial Consortium (OGC)¹⁴ Web Processing Services (WPS) (Open Geospatial Consortium, Inc., 2007b). LmRAD uses a GIS client for for web service communication and data visualization. In this way LmRAD describes the composition of natural communities using PAM's and species range maps as primary data inputs. The primary software components of LmRAD are a set of macroecology WPS services, a multi-platform Python client library for interacting with the WPS services and a Python¹⁵ plug-in for QGIS.¹⁶

The analysis of diversity patterns at bio-geographical scales, i.e. continental to global extents, significantly increases the size of the PAM when describing whole taxa over such ranges and presents computational challenges during the construction or intersection of the PAM with species inputs, randomization of the PAM, and the generation of PAM statistics of diversity and range with linear algebra operations on large matrices. PAM's are often hand constructed requiring intersecting thousands of species range maps with a data grid. Construction of PAMs can be an extremely time consuming data management task for researchers when using a heterogeneous collection of GIS tools and statistical packages each with a required learning curve.

The role of species association at larger biogeographic scales remains an important question (Arita, et al. 2008). General assembly rules of species interaction have been called into question since Connor and Simberloff (Connor and Simberloff, 1979) showed that Diamond's (Diamond, 1975) assembly rules for species association and the determination of the composition of natural communities were not different than

¹⁴<http://www.opengeospatial.org>

¹⁵<http://www.python.org>

¹⁶<http://www.qgis.org>

could be expected by chance. Since Connor and Simberloff, null model generation using randomized PAM's has become a standard but computationally intensive methodology for testing species associations and their contribution to diversity patterns. LmRAD offers two randomization algorithms. In one type of null model analysis, the matrix is randomized to produce patterns that would be expected minus prohibitive species interaction (Gotelli and Graves, 1996), while keeping species richness and range size marginal totals intact. To see why randomization of potentially sparse binary matrices can become computationally intensive you could consider the proportional fill of the matrix, p , given as the total of presence values in a matrix $\delta_{i,j}$ divided by the product of total columns S and total rows N , where, $p = \frac{\sum_i \sum_j \delta_{i,j}}{NS}$. A swap algorithm that keeps range sizes and richness totals intact must search for sub matrices of the form $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ with any consistent distance between the vertical and horizontal components, therefore the probability of finding a swap becomes $P(A) = 2p^2(1-p)^2$. So the probability for finding a swap in a random fill of $p = 0.5$ is $P(A) = 0.125$, and represents the upper limit of the probability of finding a swap (J. Soberon, personal communication, Feb. 4, 2011). More typically fills are much lower. In two examples, global terrestrial mammals and Kansas flora, the proportional fills are $p = 0.014$ and $p = 0.12$ respectively. In the first case this leads to the probability of a swap occurring once out of approximately every 2624 searches of a matrix with 6.3×10^8 elements requiring several thousand swaps for randomization. Current popular desktop software for randomizing PAMs have proven insufficient when randomizing such large PAMs (J. Soberon, personal communication, Jan. 21, 2011). The computational constraints of operating against large matrices prompted the design of LmRAD as a client-server architecture using WPS, as opposed to a strict desktop software.

To further our understanding of the software needs for macroecology we worked with ecologists from University of Kansas to answer questions pertaining to what methods were currently used for PAM construction. We sought to answer what the outstanding issues were concerning modern methods for randomizing PAMs, why computational constraints were present, what dissatisfaction existed with current software, and how best to operationalize methods for analysis of PAMs. We also sought to answer questions pertaining to specific research questions of range and diversity and how data from PAMs were best visualized. To answer these questions we began by modeling a range and diversity experiment as Python objects that could be built from information stored in a PostgreSQL¹⁷ database and exposing methods against those objects as simple WPS services. We began by attempting to adapt PyWPS¹⁸ (Cepicky and Becchi, 2007) to our current architecture but found it easier to start with the asynchronous architectural components already developed for our current Lfemapper modeling services and adapt our own WPS implementation to those specifications.

The web services approach also known as Service Oriented Architecture (SOA) provides a more 'democratic' mechanism for accessing geostatistical algorithms for ecology that would otherwise be bound within specialist GIS and spatial

statistics software. SOA promotes interoperability and loose coupling across remote systems allowing discovery and consumption of processes and chaining of those processes. By exposing spatial and statistical algorithms as Web Processing Services for generating PAMs of species data, as well as to create species and range indices and null hypothesis data inputs for these indices we are bringing together a set of tools into a single framework that will allow users to supply their own species data in addition to gaining access to species data from Lfemapper predicted range maps and use these data to populate PAMs and build range diversity plots by species and by geographic site following (Arita et al.'s 2008), methodology for the mathematics of range-diversity plots and finally to visualize these outputs across geographic and phenetic data spaces.

While there are reports in the literature about SOA and GIS with respect to WPS (Friis-Christensen et al. 2007) and others that are strict distributed GIS approaches (S.S. Wang and D.Y. Lili, 2004) fewer efforts have been detailed in the literature that address grid computing and distributed computing with GIS analysis using WPS. (Meng, Xie, and Bian, 2010, Lanig and Zipf, 2009, Muller et al. 2010) Other studies focus on WPS and its applicability to ecology, (Graul and Zipf, 2008), though the use of WPS with most ecology tools is sparse. Macroecology and biogeographic applications that have adopted WPS seem to be in their infancy with some notable exceptions. Forecasting biomes of protected areas using WPS is currently addressed by eHabitat which is conceptualized as part of the Model Web. (Dubois, et al. 2011, Skoien et al. 2011). The Model Web seeks to improve the predictive capacity of ecological models by increasing the connectivity and interoperability of models on the Web so that they can work together using web services (Geller and Turner, 2007). As part of the Model Web, eHabitat enhances its forecasting capabilities by using a discovery broker service to search for modeling services to provide inputs to other models such as climate models, to support an extended set of considerations for modeling impacts to critical habitats (Dubois, et al. 2011). Some aspects of data integration for environmental science collaboration using WPS have also been addressed by the USGS Geo Data Portal (Blodget, et. al., 2011). The Geo Data Portal allows modelers and other data users across disciplines to extract accurate spatially weighted coverage statistics from gridded data sources.

Performance issues encountered by (Friis-Christensen et al. 2007) in their description of SOA's in Spatial Data Infrastructures (SDI) stem from a fundamental limitation of such systems where the transport of data in the service chain becomes necessary if the data and the processing services are provided on different nodes in the service chain. An interesting approach for service based SDI suggests that it may be more efficient not to move the data to the processing instance, but rather move the algorithmic code closer to where the data resides when large amounts of data need to be processed, (Friis-Christensen et al. 2007, Muller et al. 2010). This involves moving away from the data driven paradigm to processing-centric infrastructures which share geoprocessing functionality through the "moving code" approach (Muller et al. 2010). Muller et al. 2010 also review Grid computing ap-

¹⁷<http://www.postgresql.org>

¹⁸<http://pywps.wald.intevation.org>

proaches for SDI's as a type of 'moving code' approach where processing tasks are passed to a middle-ware that shares the code with different nodes of the grid. This approach still has the constraint of requiring the transfer large datasets to each processing node at runtime. They conclude by pointing to the need for more research on Grid infrastructures and Geoprocessing services. In such systems a processing service interface and a deployment service provide access to the Grid environment. It would be possible to create a service based Grid, where each node would host a processing service creating a service array for parallel processing (Muller et al. 2010). Muller et al. 2010 describe an architecture following similar approaches from Friis-Christensen et al. 2007, for 'translucent' chaining of services, allowing a user to define work flows as a chain of operations and then pass this chain as a single execute request to a service instance, allowing for ad-hoc algorithm deployment.

LmRAD is being incorporated into the existing Lifemapper architecture at the Biodiversity Institute at the University of Kansas. Lifemapper is a robust archival and species distribution modeling platform consisting of a computational pipeline, specimen data archive, predicted species distribution model archive, and an Open Geospatial Consortium (OGC) and Representation State Transfer (REST) based suite of Web Services for on-demand modeling using openModeller.¹⁹ Species distribution models are created by using climate scenario data and aggregated specimen occurrences from the Global Biodiversity Information Facility (GBIF).²⁰ Existing Lifemapper computational resources provide data through web services for species occurrence, environmental data, and predicted habitat maps for various research and education laboratories.

Components that have been completed for LmRAD include web services to 1.) define and construct a gridded study area; 2.) intersect species input layers with that grid; 3.) construct a PAM from the intersected layers; 4.) compress and analyze the PAM; 5.) randomize the PAM; 6.) assess the significance of the PAM against the randomized null model; 7.) create Range Diversity plots; and 8.) a client library to interact with those services. The QGIS plug-in client library connectors, have also been completed, and services are producing preliminary plots, and the outputs can be visualized in the QGIS client. (See Figure 1.) We have implemented synchronous mode execution for smaller jobs on a web server and the processes are returning data products for continental sized PAMs at reasonable response times, while asynchronous WPS execution will be achieved through extensions to our current computational pipeline.

For our first client we chose the open source desktop GIS software, Quantum GIS (QGIS) as a WPS client (See Fig. 1). Quantum GIS was chosen because of its full featured GIS capabilities and its open and extensible architecture, but LmRAD has been built with an open, decoupled approach that should allow a variety of standards based clients, such as the scientific workflow environment software VisTrails. The QGIS client interface consists of a Python plug-in and a client library that abstracts the client functionality generalizing it for use across clients. Each Lifemapper web service is capable of self documentation by producing standard EML for re-

sulting datasets containing a process metadata section that is readable by the Lifemapper EML reader. This extended EML fully documents experiment results, allowing an experiment to be repeated across clients.

The remainder of the article will start by introducing one type of PAM analysis dealing with range-diversity plots, requiring a particular methodology for analyzing the data in the PAM, and requiring a specific method for future directions in interactive visualization of the data in the PAM that require a linked spatial representation in a GIS. We compare this to current species diversity visualization software and note that this specific type of visualization is not addressed by most species diversity software packages. This is followed by a description of the steps and elements of a range and diversity experiment and how the range and diversity data objects for an experiment that follow from those steps have been implemented in LmRAD. Methodologies for testing null hypotheses against the PAM data are described and how two of these algorithms were implemented using OpenSource libraries and were exposed as WPS services. Next, a description of our current implementation of WPS is discussed which has been designed to allow us to move to a more distributed computing approach with WPS on our compute cluster. A discussion of client architecture and some of the GIS functionality for PAM construction with QGIS is discussed. Finally the early results of testing, limitations, considerations and directions for further work are discussed.

Motivation and Methodology

Range-Diversity Plots

Most traditional analysis of PAMs focus on the distribution of species or their ranges represented by the number of sites where species occur, or on the diversity of sites, the number of species within sites. These approaches ignore the relationships between the axes of the data in the PAM. Arita et al. 2008, have shown that a link between these two variables can take into account information from both axes of the PAM, describing two different correlations between species diversity, and range size. The link is the correlation between the species diversity of sites and mean range size of species occurring in the site. A second correlation is between the range sizes of species and the mean species diversity within those ranges where the species occurs. Arita et al. 2008, show that both correlations are mirror images of the same patterns reflecting fundamental mathematical and biological relationships represented by the PAM. The per-site mean range size is a measure of a 'dispersion field' which is the set of geographic ranges of species occurring in a given site. The dispersion field volume of a site, based on the average covariance of a site with all the sites within the extent of the PAM, can be interpreted as an index of similarity between sites. Comparing the range size of a species occurring in a site is equivalent to studying the covariance of that site with all sites in the PAM. The analogue to that measure but by species is the 'diversity field' which quantifies the species diversity of all sites in which a particular species occurs. Analyzing the diversity field within the range of a species is equivalent to studying

¹⁹<http://openmodeller.sourceforge.net>

²⁰<http://www.gbif.org>

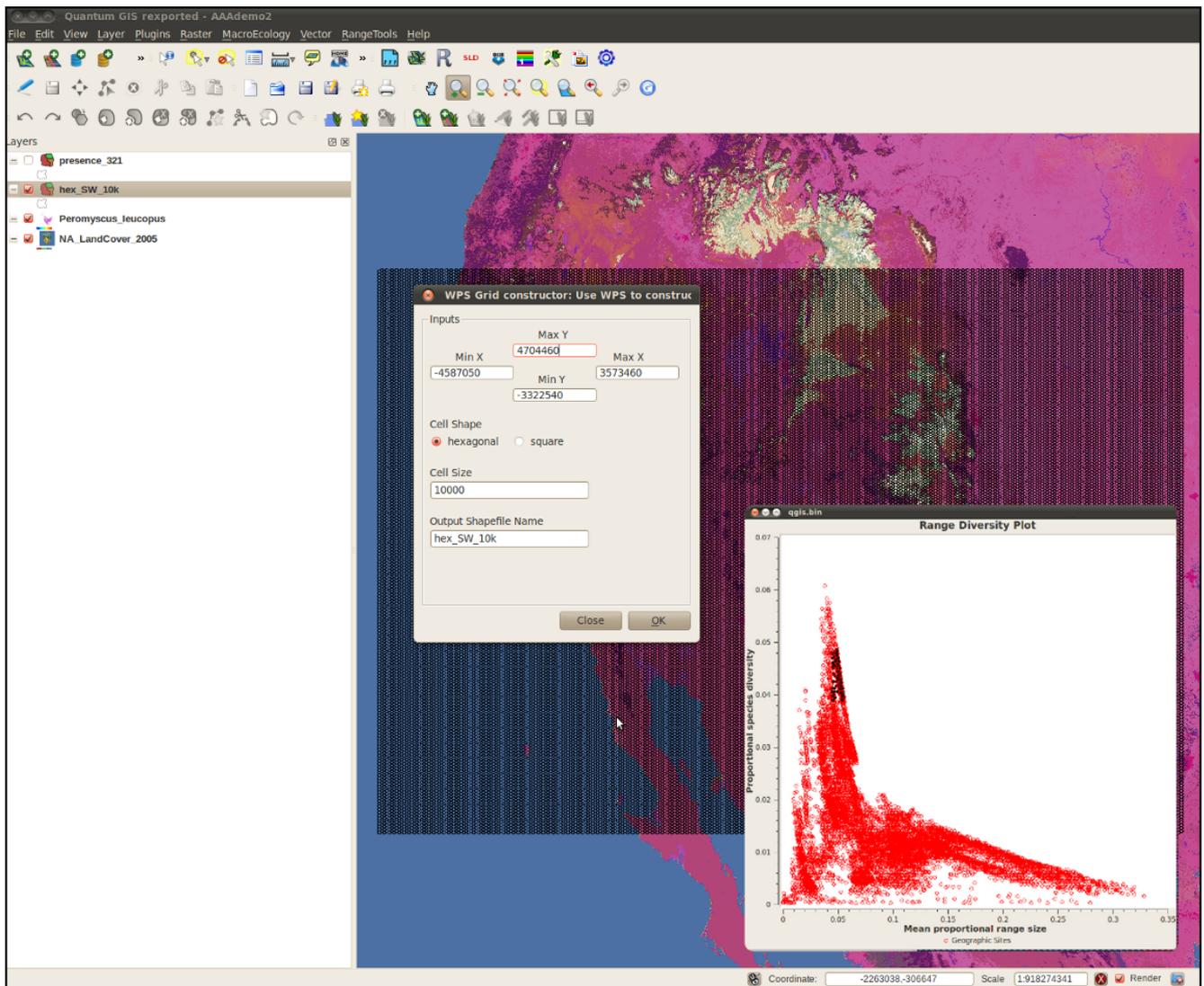


Figure 1: QGIS WPS client plug-in

its covariance with all the species. These relationships inform a "by-species" and "by-sites" view of the fundamental relationships present in the PAM. Arita et al. 2008, introduce the range-diversity plot as a way to depict species diversity of sites and geographic ranges of species simultaneously in two different types of plots. The "by-species" plot describes the relationship between the mean proportional species diversity of sites in which a particular species occurs and the proportional range size of a species. The "by-sites" plot describes the relationship between the mean proportional range size of the species within a site and the proportional species diversity of sites (Arita et al. 2008).

The visualization and analytical goal of the LmRAD platform will be to depict species diversity of sites calculated from the PAM along with species views of the PAM by using panes for different data views within QGIS. The dispersion of sites in an interactive "by-sites" range-diversity plot will be linked geographically to a map where the quadrants or

cells of the PAM are represented visually as a grid. Three data spaces will be linked and data points will be able to be selected and shown across maps, dendrograms, and plots. For species associations the data will depict association of species and link average proportional species diversity of sites containing a particular species correlated with range size in an interactive range-diversity plot with their association in a dendrogram, representing their phylogenetic relationship or morphological attributes. The range-diversity plots will be linked in the client software allowing for 'brushing' of datasets by species or location across tree data space and geographic data space, respectively. This will allow for cross-view interactivity, where selection of data points in one view of the data will highlight the data points from a different perspective in a related visualization space. (see Figure 2.)

²¹www.purl.org/biodiverse

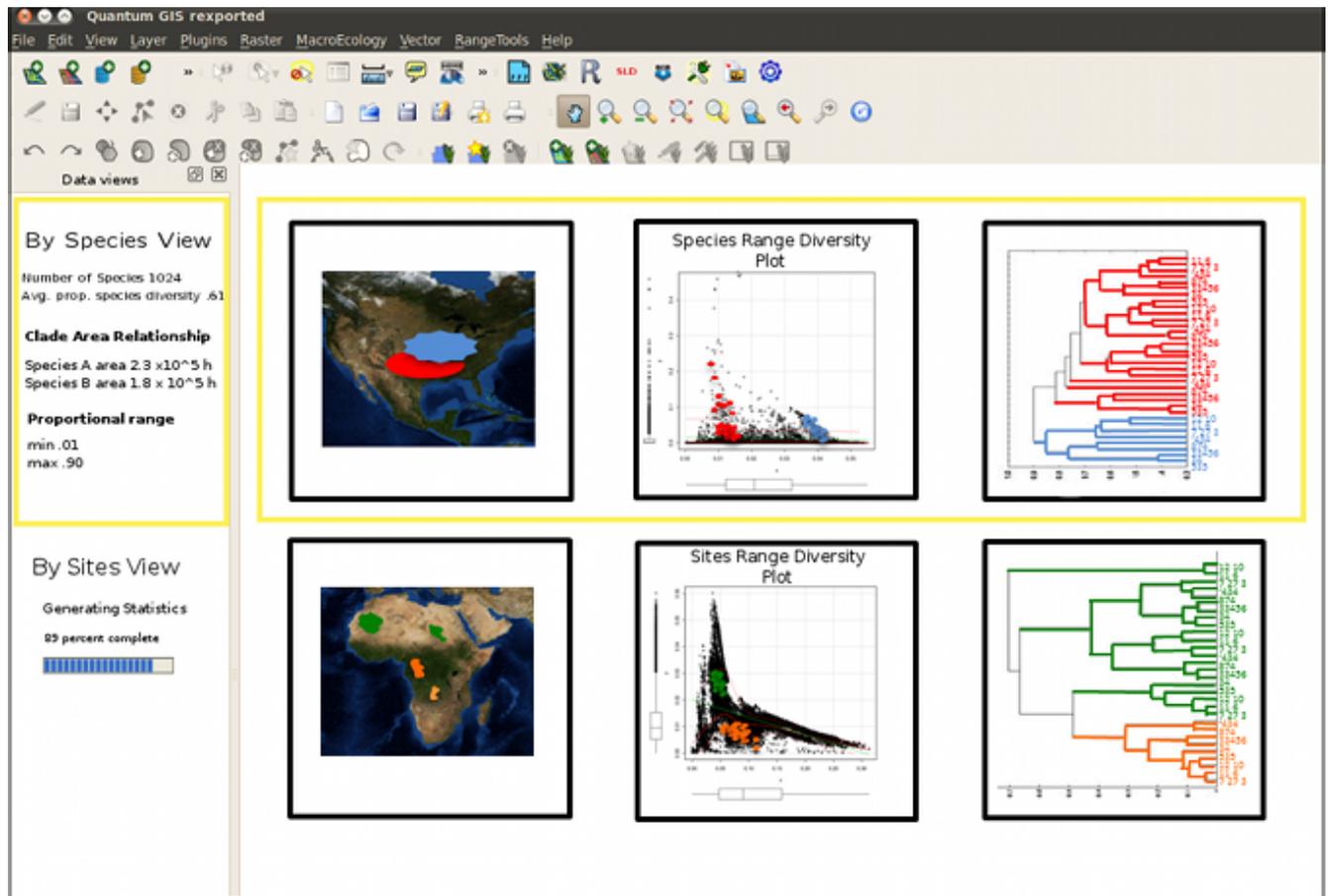


Figure 2: Proposed data panes in QGIS client showing three possible views of the data from a PAM.

Comparison of Approaches

One very interesting approach to investigating and visualizing phylogeographic relationships in a desktop software is BioDiverse²¹, (Laffan, Lubarsky, Rosauer, 2010). It provides linked visualization of data distributions in geographic, taxonomic, phylogenetic, and matrix spaces, providing the calculation of diversity indices with a focus on moving window/cluster and neighbor analysis, along with randomization for hypotheses testing. Similar to a future direction for LmRAD, Biodiverse provides tools through its windowing analysis for demonstrating the effect of scale on different diversity statistics. The set of indices that can be calculated can be customized through a scripting language. With Biodiverse, as opposed to our approach, data and results are stored in a native format and can then be exported for use in a GIS or statistical package. LmRAD starts out by extending a GIS desktop environment with statistical services and the data can then be returned and analyzed further with the existing GIS tools in QGIS.

The core difference between other software that explore phylogeographic relationships, is that LmRAD will incorporate range-diversity plots into the visualization and data space. The range-diversity plot depicts species diversity of sites and geographic ranges of species simultaneously. Phylogenetic software that have a geographic visualization space

focus on clade-area relationships, but our motivation for building a suite of tools for dealing with PAMs seeks to explore more complex mathematical and biological relationships present in the presence absence matrix, dealing with the correlations between the species diversity of sites and mean range size of species occurring in those sites and the converse relationship between range sizes of species and the mean species diversity within those ranges where the species occurs. This focus lends itself to both a "by-sites" and a "by-species" data view that can be expressed through different types of range-diversity plots, one linked to the geographic representation and the other to a phylogenetic or phenetic tree data visualization. This is more than a clade-area view of the data and represents more complex relationships of the dispersion of data points in the range-diversity-plots in terms of the degree of association among species and similarity among geographic sites.

Spatial Analysis in Macroecology, Rangel et al. 2010²² offers another software alternative for biodiversity experiments. SAM offers a comprehensive set of tools for spatial statistics, some simple mapping tools and advanced spatial auto-regression models. It uses extremely optimized linear algebra libraries for large matrix operations. The data table in SAM accommodates PAM data, and can be formatted as ESRI shapefiles. This data structure can also be used in ad-

²²SAM: <http://www.ecoevol.ufg.br/sam>

dition to PAM data to store and analyze species trait data to compute richness patterns according to different trait criteria. LmRAD uses associated data matrices to achieve the same purpose. Just as in LmRAD, data grids can be prepared in SAM at any extent and resolution as shapefiles and PAMs can be generated directly from the shapefile. Additionally, as in LmRAD, environmental raster data can be summarized over the grid cells in the data grid and added as additional variables for analysis. Also very similar to future developments in LmRAD is a tool offered by SAM called Pattern Finder that allows a user to geographically link scatter plots and maps, where grid cells can be selected in a map and then correspondingly highlighted in a scatter plot or vice versa, allowing a user to detect outliers. SAM also has sophisticated tools for evaluating the changes in spatial correlation as they are affected by changes in scale. Currently LmRAD's data model is set up to enable the user to change the scale of the PAM by taking submatrices as individual samples from an existing PAM and recalculating diversity indices, a future direction is to be able to visualize these changes. Species Distribution Modeling (SDM) is touched on in the SAM package with a routine for logistic regression, however, it does not provide many of the different algorithms used for SDM, whereas LmRAD is built on top of Lifemapper, the core functionality of which is to provide SDM algorithms as Web Services and soon to be offered as WPS services. One difference that may be pointed to here is that SAM is a Windows dependent desktop software, where LmRAD provides a cross-platform library for interfacing with remote WPS services.

EcoSim²³ is another Windows based macroecology software built specifically for dealing with null model hypotheses testing and PAM data (Gotelli, and Entsminger. 2011). It provides a variety of randomization routines for each of its modules. The co-occurrence module randomization routine allows row and column constraints, including fixed-sum similar to LmRAD. EcoSim also allows equiprobable, proportional, and weighted constraints (Ellison et al. 2000). LmRAD currently has two randomization algorithms. EcoSim has four different ways of dealing with sparse or degenerate matrices (with empty rows and columns) (Ellison et al. 2000). LmRAD currently has a compression algorithm for compressing and re-expanding such matrices, but could gain from investigating alternative approaches for dealing with degenerate matrices. The limiting factor for EcoSim seems to be the size of the PAM; 240,000 cells, or approximately 800 by 300 rows and columns is an absolute limit. One of the core requirements first addressed by LmRAD was being able to work with much larger matrices. Initial tests in LmRAD for randomization algorithms were done for matrices in the range of 6.3×10^8 cells. The computational constraints inherent in platform dependent systems and desktop software are the chief gap that LmRAD strives to solve through WPS.

User defined workflows are addressed by LmRAD through the recording of Web Service calls in a process metadata extension to the Ecological Metadata Language.²⁴ These workflows can then be re-executed in client packages incorporating the Lifemapper EML reader (Grady et al. 2011). McFerren et al. 2010 investigate a similar approach for scien-

tific workflow environments and the possibility of utilizing FOSS4G libraries for the scientific workflow environment Kepler (McFerren et al. 2010). Issues for geospatial visualization in Kepler were encountered and McFerren et al. 2010 suggest that these issues could be resolved by exporting data back out of Kepler for use in QGIS (McFerren et al. 2010). These issues may be remedied in LmRAD by either embedding QGIS libraries into a more accommodating workflow environment, currently we have embedded OpenLayers²⁵ into VisTrails²⁶, or building workflows into QGIS using the Lifemapper EML reader. McFerren et al. 2010 found that workflow environments can be freed from having to deal with library dependencies through the use of standardized web service interfaces. One implication is the need for libraries that expose complete client implementations for all of the major OGC standards. (McFerren et al. 2010). The LmRAD client library deals with much of the abstraction for bringing OGC services into both QGIS as a client, and the VisTrails environment (Grady et al. 2011).

Range and Diversity Experiments

A Lifemapper range and diversity experiment starts with the construction of a PAM. Information about species input layers for PAM construction are stored with threshold parameters for presence and absence set by the user through a REST based interface to a PostgreSQL database along with information about the study area extent, resolution, and cell shape. The PAM has two related matrices that hold intersected environmental information, species trait information, and summary statistics for each site present in the grid and for each species for use in a "by-sites" analysis. Randomized PAM's for null hypotheses tests are stored on the server-side file system in a user data space and information used for their retrieval and data about the methodology used for their randomization stored in the database. A user may collect both species data and environmental data related to the geography of the PAM, but can also use OGC services as input parameters in a WPS execute requests to provide inputs from other providers. A user's experiments can be retrieved at different stages of analysis and recreated. Provenance and repeatability of experiments will be enabled by recording service-based 'events' in process metadata within an EML document by the WPS services (Grady et al., 2011). The QGIS client and the VisTrails client will incorporate an EML reader that will then allow other scientists to be able to recreate experimental results on the same data. The Lifemapper EML reader for VisTrails is currently in early testing and the same library will be used for reconstructing workflows in QGIS.

A range and diversity experiment consists of an input layer set (both species range and environmental), a regularly spaced grid, the resulting PAM, randomized PAMs, and related environmental matrices and summary statistics matrices. A set of species input layers in the form of raster or vector species range maps are defined by the user with thresholds by for calculating presence and absence. Those data are cataloged as members of a layer set, then immediately available as OGC web services. Using these data the user is able to

²³<http://garyentsminger.com/ecosim>

²⁴EML: <http://knb.ecoinformatics.org/software/eml>

²⁵<http://openlayers.org>

²⁶<http://www.vistrails.org>

build the PAM by intersecting both species and selected environmental data layers against a geospatial grid. The PAM starts as an equal area grid defined by the user using a grid constructor WPS service. The service allows the user to define resolution, cell shape (square or hexagonal) and extent of the grid. WPS based intersection tools provide optimized intersection methods for intersecting multiple types of species and environmental inputs against the grid. Species data can be provided by the user, retrieved from repositories such as DataONE²⁷, or be chosen from a listing service that exposes species distribution model outputs from Lifemapper. Once species and environmental layers are assembled the PAM is populated with these inputs, where the presence and absence of a species are determined using the threshold parameters and are recorded as 0's and 1's in a binary matrix. A compression algorithm can be used to remove zero sum columns or rows insignificant to the matrix statistics and unnecessary for the sequential swap algorithms for randomizing the matrix. De-compression recreates the full PAM to re-randomize using different methods that require the entire geospatial extent or to add new species or environmental data to the matrix. Statistical algorithms for calculating range and diversity indices can then be performed on observed data or modelled data.

The Python NumPy²⁸ library built with the Basic Linear Algebra Subprograms (BLAS) library provides most of the matrix and statistical calculations against the PAM. Some of the diversity indices that can be derived from the presence/absence matrix include total richness or gamma diversity, local or alpha diversity, its distribution and average, and Whitaker's beta diversity (Arita et al. 2008, Borregaard and Rahbek, 2010). As a multi-species analysis covariance is central to the composition of the range-diversity plots derived from the PAM, including matrices of covariance of composition of sites, covariance of ranges of species, mean composition covariance, mean species covariance, and mean range covariance. The advantage of the PAM is that it combines information on range and diversity and co-occurrence of species measured by the degree of covariance in the matrix (Arita et al. 2008).

LmRAD will provide inputs to hypotheses dealing with assembly rules tested against species associations that keep marginal totals of occupancy and diversity intact while randomizing the presence absence data using Gotelli's recommendation for sequential swap algorithms. (Gotelli, 2001, Gotelli, 2000). Sequential swap algorithms are also the preferred method for generating null matrices in studies that analyze the degree of nestedness in a PAM. Nestedness is a structure found in the PAM where triangular submatrices of species presence 'nest' themselves into a corner of the PAM, this phenomenon, while debates range on how to measure its degree, represents the idea that less diverse niches represent subsets of more diverse niches (Brualdi and Sanderson, 1999).

A second algorithm for hypotheses-testing in LmRAD generates null models by using a dye dispersion algorithm, a 2-Dimensional geometric-constraints model that assumes range continuity. Range allocations are reassembled using a process of expansion where a species range is allowed to 'grow' to a prescribed size akin to dropping a volume of dye onto a 2-dimensional bounded surface (Jetz and Rahbek, 2001). Different methods of randomizing the PAM require

different compression states of the PAM. The swap algorithm leaves marginal totals for occupancy and diversity intact and requires searching the PAM for pairwise disjoints or checkerboard patterns with a random distance between its horizontal and vertical components. Columns or rows with no presences represent a zero sum marginal total and must be removed to increase the efficiency of the swap algorithm. This is achieved by a compression algorithm. By treating the PAM as a Boolean grid, bit-wise operations increase the efficiency of randomizing large PAMs. The Dye Dispersion algorithm for mixing of species ranges requires the PAM be uncompressed so a species range can grow within the geographic limits of the entire PAM. Statistical algorithms for calculating range and diversity indices are then performed on the randomized data.

LmRAD will provide several methods for dealing with user defined scales when constructing the PAM and comparison across PAM's. Spatial scale in PAM based studies includes both the extent of the area studied and the cell or quadrant resolution (Lira-Noreiga et al. 2007). Scale considerations are key in understanding biodiversity patterns, and are known to affect the patterns and processes that determine spatial patterns of species richness. Differences in scale directly affect estimates of average alpha diversity, gamma and beta. Whitaker's diversity measure and the fill of the matrix depend on the number of sites that are used describe the distribution of the species and therefore they are sensitive to the changes in the scale of analysis (Arita et al. 2008). LmRAD has been developed to accommodate changes in scale to enable a user to compare changes in diversity indices.

Informatics and Architecture

WPS on a compute cluster

The current Lifemapper architecture utilizes a set of RESTful web services written in Python for exposing environmental layers, modeling outputs, species occurrence points, and modeling services. When incorporating WPS into our existing architecture we wanted a Python centric solution and one that avoided Common Gateway Interface (CGI) implementations or implementations reliant on mod_Python and also one where processes can operate in a variety of different computing environments. Rather than adapt existing WPS implementations, LmRAD utilizes an autonomous job object paradigm for executing WPS requests. On the front end we use templating for document generation and a computational pipeline and PostgreSQL database to store asynchronous requests. The web architecture is generalized allowing us to switch in and out of web frameworks. The web server hosts a WPS module that registers all of the WPS services, exposes metadata classes for each one, and handles requests to individual processes, including GetCapabilities requests which return a description of all available processes. A metadata service module for each process registers inputs and outputs for the process, handles the DescribeProcess request and exposes an execute method. The execute method is responsible for entering the job into the database. Then the computational pipeline picks up the job, assembles a job object, serializes it and posts it to the head node of the cluster. On the cluster,

²⁷<http://www.dataone.org>

²⁸<http://numpy.scipy.org/>

the job is de-serialized and pickled so that the target node can request the job object from a web server on the head node. Worker threads simultaneously update experiment status and inputs, submit experiments to and retrieve results from the compute cluster. The head node of the cluster submits the job ID and job Type, and a pipeline ID to the Sun Grid Engine scheduler. The scheduler farms out the job to a node where a job factory initializes the appropriate job runner with the appropriate environment variables using classes of functions specific to the environment. One of these will be passed to a job runner when it is instantiated so that it knows where to store data, how to access data, and how to update the status. Job input requests for raster data for species layers are requested by the node using OGC Web Coverage Service (WCS) requests from mapping services that use MapServer (<http://mapserver.org/>) against the species distribution model archive in Lifemapper.

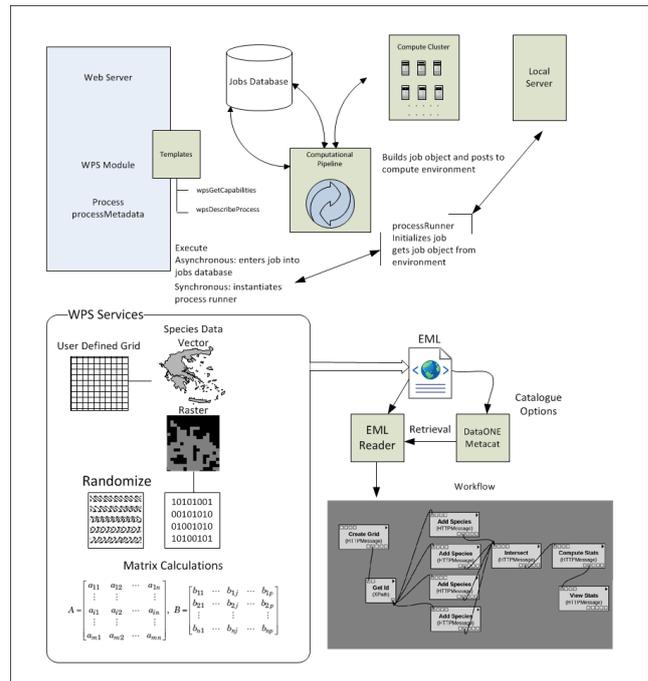


Figure 3: Overview of WPS and cluster architecture.

GIS operations, visualization and client architecture

The work performed by our WPS processes is done by a job runner. These job runners are not aware of the environment where they physically exist. This is possible by abstracting the methods used to interact with their environment. When the job runner is instantiated, it is given a class that includes methods for interacting with the environment. These environment methods implement a standard interface for interaction with the job runner. The job runner expects a standardized object of a particular type from each method. If the environment is one of our core machines, the job object might be constructed by accessing the database and building the object from that response. On a cluster node, an HTTP request is sent to the front end of the cluster that sends back a serialized version of the object that is in-turn de-serialized and returned to the job runner. In the testing environment, an XML file is accessed on the file system and then restored to an object. Each environment can perform the tasks through different methods as long as the post condition is the expected object. This also applies to other methods, e.g. methods that return a file path for writing temporary data or updating the job's metadata. The job runner is decoupled from its environment, which allows the jobs to be agnostic of where they run and also allows them to be flexible enough to move. To move a job to a different environment only the registration needs to change. If a job type is found to be too resource intensive on the core machines, it can be registered to run on the cluster and the performance bottleneck can be alleviated quickly and simply. Figure 3 outlines the LmRAD architecture.

GIS operations are exposed as WPS processes through a set of dialogues within a Python plug-in for QGIS. The plug-in can be used for building the data grid for the PAM and intersecting species and environment layers and returning PAM statistics. Geometries for the cells in the PAM created as a shapefile using the OSGeo OGR²⁹ library and returned by the WPS service to QGIS for visualization. The shapefile is also stored on the processing node of the service instance for use in further processing. A user can then choose from existing predicted species distribution maps in Lifemapper provided by a listing service, or upload raster or vector species maps. The presence and absence of a species is then calculated by intersecting the grid with the species layer and that data recorded in a Numpy matrix, using the regular grid as a place holder for the geographic data. The intersection algorithms use R-tree spatial indexing³⁰, matplotlib³¹, and NumPy matrix manipulations with the OSGeo GDAL (GDAL 2010) and OGR libraries to achieve efficiency and speed of operation. The user may expand or reduce the study area and add or subtract species columns. Visualizations of these changes will provide some challenges and further work must be done to bring these into QGIS.

The QGIS client architecture uses a threaded architecture that uses the PyQt widget toolkit³² with signals and slots to implement the Observer pattern (Gamma, et al. 1995) in order to handle asynchronous WPS requests. The Lifemapper client library abstracts the communication layer away from specific model implementations for a client. A model (in Mod-

²⁹<http://www.gdal.org/ogr>

³⁰<http://pypi.python.org/pypi/Rtree>

³¹<http://matplotlib.sourceforge.net>

³²<http://www.riverbankcomputing.co.uk/software/pyqt/intro>

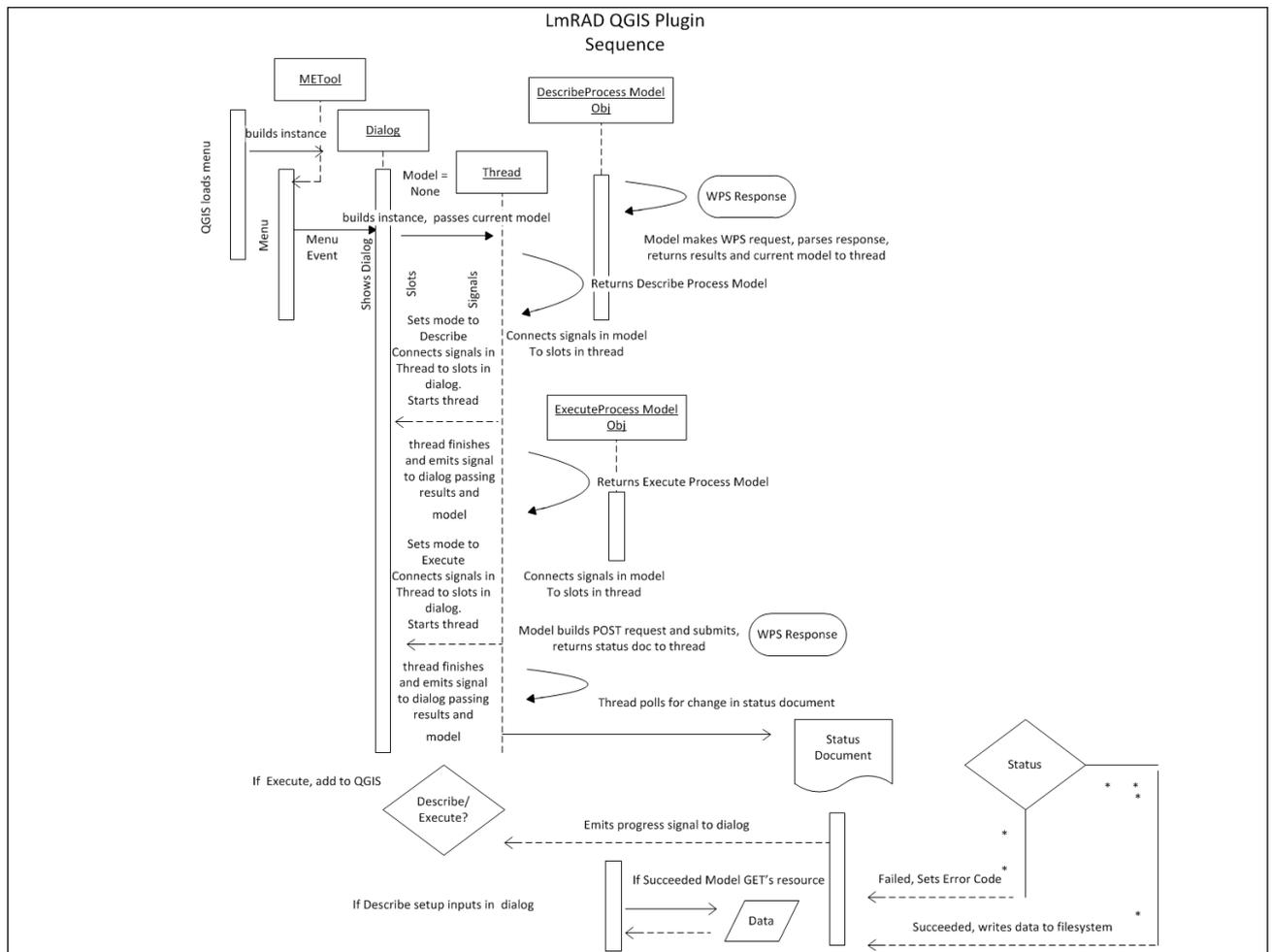


Figure 4: LmRAD QGIS Sequence Diagram for asynchronous requests

el/View/Controller) consists of a de-serialized Python object derived from the WPS XML, constructed using a recursive method in a wrapper to the Element Tree Python library. All dialogues for the suite of range and diversity tools inherit from a common set of dialogue classes, allowing different dialogues to be spun up readily. Figure 4 describes an asynchronous request from the QGIS client.

The Lifemapper client library, used for both LmSDM modeling and LmRAD experiments, is a Python module designed to interact with multiple clients to minimize code duplication. It communicates with the Lifemapper web services, both RESTful and OGC compliant. The library can be initialized as an anonymous user so that only public data from the Lifemapper web services may be accessed, or, the client may be initialized with user credentials allowing access to a private Lifemapper workspace. The client abstracts details of the exact interface to services and provides for a simpler interaction with the services for the developer. Hiding this underlying complexity does two things, first, application developers using the client library do not need to know the

details of different communication standards, and second, implementations can change without requiring modification to end-user software.

Statistical operations are exposed as WPS services through the plug-in in QGIS. Once an experiment has been setup to run using a constructed PAM, statistical services can be used to retrieve range and diversity indices derived from the PAM. Range-diversity plots are linked in QGIS using PyQt Widgets for Technical Applications (PyQt³³) allowing for 'brushing' of datasets by species or location across geographic space and in plot space and in the future, tree data space. Two different types of range-diversity plots map the relationships between the mean proportional species diversity against the proportional range size of a species; and the relationship between the mean proportional range size and the proportional species diversity. These two different 'views' of the data can then be linked for display in dendrograms in the case of the species specific plot and geographic space in the case of the locality specific plot.

³³<http://pyqt.sourceforge.net>

Findings, Considerations and Future Directions

This article has demonstrated the benefits of a distributed WPS approach within an SDI for macroecology tools. We have described our current efforts based on the needs described for working with PAM data which have become very large in current macroecology research efforts. We have addressed a need discussed in some of the literature (McFerren et al., 2010) for a library that can be used both from within QGIS and a scientific workflow environment with a common API for working with OGC standards. By building on an existing GIS platform we bring a more powerful set of tools to bear on macroecological problems and geospatially enable the scientific workflow environment software VisTrails, side stepping issues dealing with conflicting libraries by using WPS services.

Early results of this methodology have been very promising. Testing of the swap randomization algorithm on PAM matrices with cells and a very sparse fill provides one solution with 30,000 random searches for 15 suitable submatrices in less than one second. Increasing the search to 600,000 increases the number of swaps to about 155, with one solution found in 3.8 seconds. A more typical number of swaps is achieved with 12,000,000 searches on the dataset, with 3,617 submatrices found in 77.58 seconds. The vectors for the range-diversity plots are calculated against the same matrix in less than 3 seconds. Basic diversity indices are also being calculated but more computationally intensive matrix algebra algorithms still need to be developed and will benefit from being farmed out to the cluster environment. The intersection algorithms have also proved to be efficient. The current Python client library has been in use with the Lifemapper REST and OGC services for some time and has proved reliable. The implementation adjusted for WPS should prove to have the same reliability. Early testing on the library has been successful, and a future direction for it is to have it dynamically add functionality based on an online configuration document describing the services available in the Lifemapper system.

Deeper geostatistical functionality still needs to be developed. Future directions include interactive visualization for dendrograms or phentic trees in QGIS, along with visualization methods for quickly comparing changes in scale of the PAM. The methodology for handling asynchronous requests for WPS is not fully integrated into our computational pipeline. Range and Diversity experiments have been completely modeled in our database and the REST services retooled to account for WPS. The new WPS module has been used for range randomization in synchronous mode. Each compute environment will have its own computational pipeline, depending on the type of job, asynchronous jobs can be handled locally or on our compute cluster, but the WPS job object will be able to work in either environment, and we stress the agnostic nature of the WPS job for future implementation in a variety of environments. Work needs to be done to address the orchestration of jobs on the cluster which is currently handled in the database, this may require further research into WPS choreography or orchestration. A good deal of literature exists describing different approaches to WPS orchestration for chaining or nesting services and some within

Grid environments (Muller et al., 2010, Friis-Christensen et al. 2007, Meng et al. 2010, Diaz et al. 2010, Lanig and Zipf, 2009). Currently we leverage the high-throughput capacity of our compute cluster where one Lifemapper job runs on one node. Where possible jobs will be parallelized to run on multiple nodes using the Google MapReduce paradigm. Nodes on our cluster still must request data for potentially thousands of species range layers that can be pulled from repositories like DataONE, or our archives, but those data may also reside with a user, requiring them to upload all of their data inputs for the PAM. This represents a data-centric approach where data must be moved to the processing environment. The cluster environment provides future challenges and increased performance, that would allow us to address more of a 'moving code' approach described by Muller et al. 2010 to address the data access issue.

A different approach may prove beneficial if as summarized by Muller et al. 2010, algorithms frequently evolve, identical algorithms are shared among several service instances, and data can be shipped prior to execution, and algorithms have to be hosted at service instances close to the data to decrease bandwidth impact. (Muller et al. 2010). Data driven approaches are reasonable if one-time assembly and execution of workflows is the norm, and real-time response for complex service chains is not required (Muller et al. 2010). Michael and Ames, 2007 evaluate WPS for use within client-side GIS, and address similar concerns about moving data. They point out that WPS should not be used when operations on data can be completed more quickly locally than remotely, especially when factoring in the time to upload data, and download results. Conversely data should be sent to servers hosting service instances that have higher processing power when the time to process the data locally would be greater than the combined time to transmit the data. Similar to our approach they also note that building a WPS client into an existing GIS package takes advantage of the visualization and other GIS functionalities of the GIS. (Micheal and Ames, 2007). We were originally presented with performance issues with desktop software for operating against large PAMs. The processing time for individual PAMs fit the criteria for moving the data to a server with higher processing power. However, the data can also be local, and some outputs are best visualized and manipulated better in the local GIS client. Clearly a compromise has to be struck between the two approaches.

By employing WPS services with an established and extensible open source GIS platform we are beginning to provide intuitive, efficient and data rich biogeography tools for formulating biogeographical hypotheses and for simultaneously analyzing and visualizing species ranges and geospatial biological diversity patterns through range-diversity plots. A distributed approach for using WPS in different environments will allow current research practices of PAM assembly overcome large computational constraints and advance biodiversity research and education of the global impacts of climate change on species and biological communities.

Acknowledgements

We gratefully acknowledge the University of Connecticut (R. Colwell and T. Rangel), with co-funding from U.S. National Science Foundation Award BIO/DBI 0851290 and in

collaboration with Jorge Soberon, Narayani Barve and Andres Lira-Noreiga.

References

1. Arita, H.T., Christen, J.A., Rodriguez, P., Soberon, J. 2008. 'Species diversity and distribution in presence-absence matrices: mathematical relationships and biological implications.' *Am. Nat.* 172: 519-532.
2. Borregaard, M. K. and C. Rahbek, 2010. Dispersion fields, diversity fields and null models: uniting range sizes and species richness. *Ecography* 000: 000-000.
3. Brualdi, Richard A. and Sanderson, James G. 1999. 'Nested species subsets, gaps and discrepancy.' *Oecologia*, 119:256-264.
4. Connor, E. F., and D. Simberloff. 1979. 'The assembly of species communities: chance or competition?' *Ecology* 60:1132-1140.
5. Cepicky, J. Becchi, L. 2007. 'Geospatial processing via Internet on remote servers – PyWPS, OSGeo Journal 1 (May 2007): 5p
6. Diamond, J. M. 1975. 'Assembly of species communities.' in M.L. Cody and J.M. Diamond, (eds.), *Ecology and evolution of communities*. Belknap, Cambridge, MA.
7. Diaz, L., Pepe, M., Granell, C., Carrara, P., Rampini, A. 2010. 'Developing and Chaining Web Processing Services for Hydrological Models' in Commission IV, WG IV/5: WebMGS 2010.
8. Dubois, G., Skoien, J.O., de Jesus, J., Peedel, A., Hartley, A., Nativi, S., Santoro, M., Geller, G. 2011. 'eHabitat: a Contribution to the Model Web for Habitat Assessments and Ecological Forecasting.' in :Submitted to the 34th International Symposium on Remote Sensing of Environment, April 10-15, Sydney, Australia.
9. Ellison, A. A. and Aaron M. Ellison. 2000 Reviewed work(s): 'EcoSim: Null Models Software for Ecology.' *Bulletin of the Ecological Society of America*. Vol. 81, No. 2 (Apr., 2000), pp. 125-127
10. Friis-Christensen, Ostlander, Lutz, Michael, Bernard, Lars. 2007. 'Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions.' *Transactions in GIS*, Vol. 11, No. 6. pp. 799-81
11. Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. ISBN 0-201-63361-2.
12. GDAL, 2010 – Geospatial Data Abstraction Library: Version 1.6.3., Open Source Geospatial Foundation, <http://gdal.osgeo.org>.
13. Geller, G.N., Turner, W. 2007. 'The Model Web: a concept for ecological forecasting'. *Geoscience and Remote Sensing Symposium, IGARSS IEEE International*, 2469-2472, 23-28 July 2007.
14. Gotelli, N. J. 2000. 'Null model analysis of species co-occurrence patterns.' *Ecology* 81: 2606-2621.
15. Gotelli, N. J. 2001. 'Swap and fill algorithms in null model analysis: rethinking the knight's tour.' *Oecologia* 129:281-291.
16. Gotelli, N. J., and G. R. Graves. 1996. 'Null models in ecology'. *Smithsonian Institution*, Washington , DC
17. Simpson, G. G. 1964. 'Species density of North American recent Mammals'. *Systemic Zoology* 12:57-73.
18. Gotelli, N.J. and G.L. Entsminger. 2011. *EcoSim: Null models software for ecology. Version 7*. Acquired Intelligence Inc. & Kesey-Bear. Jericho, VT 05465. <http://garyentsminger.com/ecosim.htm>.
19. Grady, C.J., Beach, J., Cavner, J., Stewart, A. 2011. 'Lifemapper, VisTrails and EML: Documented, Re-executable Species Distribution Models'. in: *Proceedings of the Environmental Information Management Conference 2011 (EIM 2011)*, eds. M.B. Jones, C. Gries.
20. Graul, C., Zipf, A. 2008. 'Putting Biogeographic Research on the Spatial Web: Towards Interoperable Analysis Tools for Global Change Based on the Web Processing Service (WPS)'. *Digital Earth Summit on Geoinformatics: Tools for Global Change Research*. Potsdam, Germany.
21. Jetz, W. and Rahbek, C. 2001. 'Geometric constraints explain much of the species richness pattern in African birds.' *Proc. Nat. Acad. Sci. USA* 98:5661-5666.
22. Laffan, Shawn W. Lubarsky, Eugene and Rosauer, Dan F. 2010. 'Biodiverse, a tool for the spatial analysis of biological and related diversity.' *Ecography* 33:643-647
23. Lanig, S., Zipf, A. 2009. 'Interoperable Processing of Digital Elevation Models in Grid Infrastructures' in *Earth Sci Inform* 2: 107-116
24. Lira-Noreiga, A., Soberon, J., Navarro-Siguenza, A. G., Nakazawa, Y. and Peterson, A. Townsend. 2007. 'Scale dependency of diversity components estimated from primary biodiversity data and distribution maps.' *Diversity and Distributions*. 13, 185-195.
25. McFerren, G., van Zyl, T., Vahed, A. 2010. 'FOSS Geospatial Libraries In Scientific Workflow Environments: Experiences and Directions.' *Applied Geomatics*. 21 July 2010
26. Meng, Xiaoliang , Xie, Yichun, and Bian, Fuling. 2010. 'Distributed Geospatial Analysis through Web Processing Service: A Case Study of Earthquake Disaster Assessment.' *Journal of Software*, Vol. 5, No. 6, June
27. Micahel, Christopher, and Ames, Daniel P. 2007. 'Evaluation of the OGC Web Processing Service for Use in a Client-Side GIS.' *OSGeo Journal* Vol 1. May 2007
28. OGC 2007b *OpenGIS Web Processing Service, Version 1.0.0*. Wayland, MA, OGC Document No 05-007r7
29. Rangel, Thiago F., Diniz-Filho, Alexandre F. and Bini, Luis Mauricio. 2010. 'SAM: a comprehensive application for Spatial Analysis in Macroecology.' *Ecography*. 33: 46-50.
30. Skoien, J.O., Dubois, G., De Jesus, J. 2011. 'Forecasting Biomes of Protected Areas'. *Procedia Environmental Sciences* 4 (2011) 44-49.
31. Wang, S.S. and Lilu, D.Y. 2004. "Spatial Query Pre-Processing in Distributed GIS" in *Grid and Cooperative Computing*, – GCC, H. Jin, Y. Pan, N.Xiao, and J.Sun (Eds.), (Berlin; Springer) p.p. 734-744.

Jeffery A. Cavner
 Software Developer
 KU Biodiversity Institute
 1345 Jayhawk Blvd.
 Lawrence, KS 66045 U.S.A.
jcavner@ku.edu

Using GRASS and R for Landscape Regionalization through PAM Cluster Analysis

Allan D. Hollander

Abstract

Landscape regionalization is a frequently encountered need in the geographical sciences, having applications ranging from sampling design to conservation prioritization. One technique for partitioning the landscape is to use cluster analysis of GIS layers describing the area under study. Here I present a GIS technique that uses partitioning around medoids as its clustering algorithm. Partitioning around medoids (PAM) is a non-hierarchical clustering algorithm that is related to the commonly-used k-means clustering technique. PAM differs from the k-means algorithm in that a) PAM assigns cluster centroids to actual data observations, rather than using values averaged over subsets of the entire dataset and b) PAM accepts categorical data as input in addition to numerical data. These properties of PAM make it useful for landscape regionalization because often one wishes to incorporate categorical variables such as vegetation class or soil types in the regionalization. I illustrate the PAM technique with an example of sampling design for a local-scale analysis of agroecosystems in Northern California. For this work I use GRASS and R, generating in GRASS a set of random points covering the study area and attributing these points with values from raster and vector layers of interest, importing this data table into R for the PAM cluster analysis, and exporting the resulting clusters back to GRASS for geographic visualization.

Background

A long-standing concern in geography has been regionalization, or how to define areas in space that share similar characteristics according to some theme of interest. Quantitative work on regionalization methods has been undertaken for a considerable period of time (e.g. [Spence and Taylor \(1970\)](#), [Johnston \(1970\)](#) and continues to the present (see [Duque et al. \(2007\)](#) for a recent review). In the earth and environmental sciences, regionalization finds application in areas as diverse as hydrology ([Wiltshire \(1986\)](#)), biogeography (e.g. [Patten and Smith-Patten \(2008\)](#), [Procheş \(2005\)](#)), and climatology ([Stooksbury and Michaels \(1991\)](#)). Although no quantitative method can provide a single definitive regionalization suitable for all applications, it is useful to have a suite of formal methods available when a problem calling for defining a new set of regions arises. In what follows I present a simple method for generating regions for use in a GIS analysis, working with the open source GIS software GRASS ([Team \(2009a\)](#)) and the statistical package R ([Team \(2009b\)](#)).

The particular focus of this approach is to use cluster analysis methods to support studies in landscape ecology and in conservation. In conservation, one class of problems that

can be addressed by cluster analyses is reserve design, where a classical problem is to ensure that a reserve network contains sufficient representation of all elements of conservation interest (e.g. [Vane-Wright et al. \(1991\)](#)). A cluster analysis approach can serve to capture dissimilarities in the environment by placing these in different clusters and hence provide a means to adequately represent the biological diversity of a landscape (e.g. [Trakhtenbrot and Kadmon \(2005\)](#)). Another class of problems is sampling design. In ecological inventories, the number of sampling plots one can place is always quite limited. A usual strategy to increase the efficiency of one's sampling process is stratified design: that is, rather than placing sampling points randomly across the study region, one partitions the landscape across environmental variables of interest, and subsequently places similar numbers of sampling points within each partition. Cluster analysis lends itself well to defining the strata, since it provides a formal method to separate the variability in the environment into different groups.

Cluster analysis algorithms can be divided into hierarchical algorithms and partitional algorithms ([Kaufman and Rousseeuw \(2005\)](#)). In hierarchical algorithms, clusters are constructed sequentially to form a nested hierarchy of cluster types. In partitional algorithms, the set of clusters is constructed all at once, often requiring establishment of the desired number of clusters beforehand. In general, cluster analysis proceeds in two steps, the first being creation of a dissimilarity matrix, where one uses a metric to compute the distance in attribute space between a pair of observations and assigns this value to the element of the matrix where its row and column is indexed by the id numbers of the observation pair. The second step is to take the distances in the dissimilarity matrix and apply an iterative algorithm to sort the observations into different clusters.

In landscape analysis, often the environmental properties of interest are categorical variables. Examples of these include habitat types or soil classifications. The fact that some of the variables are categorical poses problems for many clustering algorithms, for instance the most commonly-used partitional clustering algorithm, k-means clustering. In k-means clustering, the number of clusters is set before running the algorithm, and the algorithm is initialized by randomly placing that number of cluster seeds among the multidimensional environmental attribute space. The algorithm then adjusts the position of these cluster seeds so as to minimize the dissimilarities within a cluster and maximize the dissimilarities between clusters. The k-means algorithm works only with continuous variables, and the cluster centroids are defined to be the mean values of the attributes of all the observations in the cluster. Since this algorithm relies upon computation of mean values, it fails to work with categorical variables, and an alternative algorithmic strategy needs to be taken if one wishes to incorporate categorical variables.

There are two elements in this strategy. The first is to use a dissimilarity metric that works with both categorical and numerical variables. One such metric is Gower's similarity coefficient (Gower (1971)). This coefficient computes the distance between two observations by averaging a difference metric over both the numerical variables and the categorical variables. This difference metric ranges from 0 to 1 for both types of variables. For the numerical variables, the difference metric is the absolute value of the ratio of the difference of the values of the two observations and the range of the variable across all observations. For categorical variables, the difference metric is 0 if the two observations share the same categorical value and 1 if they differ. The second element in this strategy is using a clustering algorithm that works with mixed variables. The algorithm used here is partitioning around medoids (PAM) (Kaufman and Rousseeuw (2005)). This algorithm is similar to k-means clustering but differs from it in that the cluster centers (the "medoids") are actual observation points rather than centroids averaged over a number of observations. The medoids therefore provide the sample point that is the exemplar for each cluster.

This approach can easily be carried out using the open source software programs R and GRASS. R is excellent for testing novel statistical techniques since academic statisticians nowadays develop many new algorithms in the R statistical environment. GRASS provides excellent integration of raster and vector GIS capabilities and scripting abilities.

Methodology

Figure 1 illustrates the workflow in this analysis. The diagram at upper left shows the stack of GIS layers together with the random points to be overlaid on the GIS stack. This creates a data table which is fed into the R statistical environment at upper right. At lower right the clusters are constructed, which are then mapped in the GIS at lower right. Below I detail the steps one takes to carry out this workflow.

1. The first step is to assemble the data layers to be used in the analysis in GRASS. Both raster and vector data can be used. It is important to recognize which datasets are categorical (e.g. land cover) and which are numeric (e.g. soil percent organic matter content), since treatment of these will vary in the statistical analysis. For vector data there may be several fields of interest within the attribute table of a single data layer.
2. The sampling density needs to be determined. There are no fast rules for doing this, but one needs to pay attention to the spatial resolution of the datasets being queried. Given a range of spatial resolutions of the datasets, there is no point in sampling at the finest resolution, but sampling only at the coarsest resolution will by definition lose a lot of the detail in the rest of the datasets. A choice somewhere in the middle of the range may be the most satisfying choice. For instance, in an application where many of the data variables come from a 1-kilometer gridded soils map, one might choose a sampling density of 4 points per square kilometer.
3. The routine for generating random points in GRASS, `v.random`, asks for a fixed number of points to be generated within the current geographic region. Computing that value from a desired sampling density requires calculating the area of the region from its bounding box and dividing through by the sampling density.
4. Often one wishes to restrict sampling to a subregion within the rectangular geographic region. In GRASS this may be accomplished by defining the subregion as a vector polygon layer, and then selecting the points that fall within that polygon using the `v.select` command.
5. Next one makes a list of the variables to be sampled and adds these as columns to the attribute table of the sample point layer using the command `v.addtable`, paying attention to the data types of the variables when one creates the columns.
6. These columns then get instantiated with data from the layers of interest, using the commands `v.what.rast` and `v.what.vect`. Running these commands over a large stack of layers can get tedious but this routine can easily be scripted.
7. The attribute table for the sample points is then exported so that it can be read into the R statistical environment. There are several options for export formats; of these a good choice is dbf since it directly preserves data type information of the data table. To export dbf, one selects the dbf option from the `db.out.ogr` command.
8. One then switches over to the the R environment. Though support for spatial datasets in R is well-developed,³⁴ which includes a direct interface to GRASS through the `spgrass6` package), the steps that follow do not depend on this interface, and hence one does not need to understand how R treats spatial data in this workflow. The first step in the R environment is to read the sample point data using the `read.dbf()` function which is part of the foreign package.
9. One then checks to see that the variables in the data frame have the correct data type. Categorical variables may have been represented as integers in this table, especially if they have been sampled from a raster, for example one displaying discrete land cover classes. Such variables need to be reassigned to categorical type using the `factor()` function in R. Conversely, sometimes string variables in the data table actually represent ordinal values, one example being soil drainage being labelled as "poorly drained", "well-drained", or "excessively drained". For analysis, these variables should be manually reassigned to integer ranking.
10. When one reads in the sample point data table into R with the `read.dbf()` function, the first column of the data frame contains the category id value from the vector sample point layer. This id value obviously is not useful in multivariate cluster analysis and needs to be omitted. This is easily accomplished by constructing a new data frame from the original data frame (function `data.frame()`), leaving out the category id variable as well as any other variables that one now wishes to omit. One then assigns the category id values to the row names with the `row.names()` function to preserve identification of the sample points through the cluster analysis.
11. There are two steps to the cluster analysis, creating a dissimilarity matrix and then running a clustering algorithm on the resulting matrix. In R, the library that does both functions is `cluster`.

³⁴See <http://cran.r-project.org/web/views/Spatial.html>

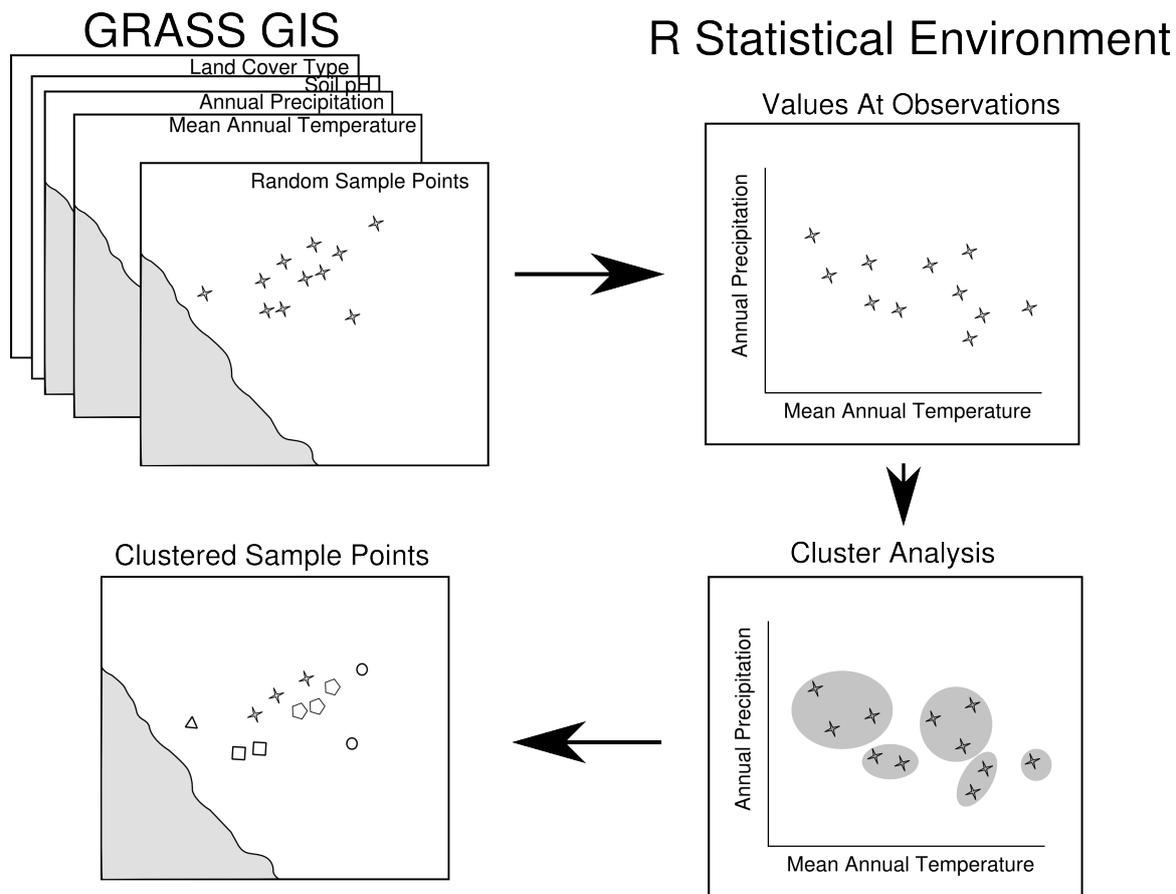


Figure 1: PAM cluster workflow

12. The routine to construct the dissimilarity matrix is called `daisy()`. It has options for several dissimilarity metrics but defaults to using Gower’s similarity metric if the inputted data frame has both numeric and categorical variables. Applying `daisy()` to the data frame above creates a dissimilarity matrix where each entry represents the environmental attribute dissimilarity between a pair of sample points.
13. This dissimilarity matrix is then fed into the `pam()` clustering routine. A key parameter that is also input into `pam()` is the number of clusters to be generated by the algorithm. Because in this set of studies the clusters were to be used in a sampling design containing relatively few plots, few clusters were desired, so the starting point for this parameter was 5 to 7.
14. The R cluster library provides a visualization aid called silhouette plotting (Rousseeuw (1987)) in the function `silhouette()`. This tool helps identify clusters which are poorly defined in the sense that many of their members could easily be placed in another cluster on the basis of dissimilarity distance. By using this tool while iterating through a broad range of numbers of clusters in a set of `pam()` runs, one can identify the choice of the number of clusters that provides for best cluster discrimination. This choice of the parameter is then used in selecting the final cluster analysis.
15. Because the PAM algorithm selects particular sample points that are most representative of each cluster (the “medoids”) it is easy to inspect the values at these sample points by extracting rows from the original data frame by the medoid id number. One can then give the different clusters interpretive labels e.g. “Mixed hardwood upland”.
16. Next one brings the result of the cluster analysis back into GRASS. There are several steps to this process. First, one exports a table from R giving the id values for the sample points in one column and the cluster analysis assignments in the second column. (The function `write.csv()` is useful for this). Next one needs to assign the cluster labels to the sample points. This is most easily done by adding a column to the sample point vector layer with the command `v.db.addcol` to contain the integer cluster labels. This column can then be filled with the cluster labels by running a series of update SQL commands that assign an integer cluster label based upon the category id value of the sample points. Shell commands such as `awk` and `sed` can be used to generate a file containing the update SQL commands from the R export table.
17. The original sample points can then be plotted on a map labeled with the cluster id values. Depending upon the application, it may be desirable to assign cluster values to the entire surface of the map region, and not just refer to the labeled sample points. One way to accomplish this is to construct a Voronoi tessellation using the command `v.voronoi` and assign each of the demarcated polygons the cluster value of its enclosed sample point.

Case Study

As an illustration of this method I present an example from a study of carbon storage in a mixed agricultural and natural landscape in Northern California (Williams et al. (2011)). Estimation of carbon stocks on the landscape has become an important concern in light of a desire to mitigate greenhouse gas emissions through carbon sequestration (e.g. van Kooten (2009)). Methodologies for estimating carbon stocks are only just now being developed, an example being the California Climate Action Registry Forest Protocol (Reserve (2009)), and these methodologies need to be expanded to a broader suite of landscapes. This study examined a set of vine tracts and the adjoining woodland ecosystems from an organic vineyard in Mendocino County (the Bonterra label of the Fetzer Vineyards). The aim of the study was to estimate carbon stocks across both the vineyards and wildlands using field sampling and model building, to compare the aboveground stocks with the soil carbon stocks, and to develop a methodology that can use these models across a broader landscape for accurate carbon accounting.

Over the 1149 hectare landscape of the study area, it was only feasible to place 93 vegetation plots and dig 44 soil pits, and to maximize the efficiency of placing these samples, we followed the PAM cluster analysis methodology outlined above to classify the landscape for sampling. In this procedure, we used four different data layers in the GIS data stack. These included vegetation type — a categorical variable describing the land cover using one of nine types in the California Wildlife Habitat Relationships system, elevation, slope, and clear-sky solar radiation summed over the entire year as modeled from the elevation map. (calculated using the `r.sun` routine in GRASS). We sampled these four layers using 1013 points distributed randomly over the Fetzer lands. With the data matrix from this random sampling, we produced a matrix giving the dissimilarity between each of the points. This dissimilarity matrix, computed using Gower’s coefficient, was used as the input into the PAM clustering routine. Based on the distinctiveness of the groupings produced when several different numbers of clusters were requested, we selected seven as the final number of clusters. Table 1 presents the values of these environmental variables for the 7 medoid points. These clusters could be characterized as following: two vineyard types, one on flat ground at low elevations and one upland type; an annual grassland type at moderate elevation; a mixed hardwood-conifer type at relatively high elevation and steep slopes; and two mixed hardwood types, one at relatively high elevation, steep slopes, and lower solar radiation, and the other at more moderate slopes and elevation. Figure 2 plots the location of the different clusters on the 5 tracts of the Fetzer lands.

Table 1 also shows the number of sample points that were assigned to each cluster. As is evident, the clustering was only used as a rough guide to distributing cluster points. No vegetation plots were assigned to vineyards or annual grasslands, since the aim was to focus on woody vegetation types. The soil pits were divided roughly equally into pits on vineyards and pits on wildlands, but the wildland pits were predominantly assigned to the Mixed Hardwoods II class. The vegetation plots and soil pits were assigned to the cluster types in the field by reference to a map of the cluster points similar to Figure 2.

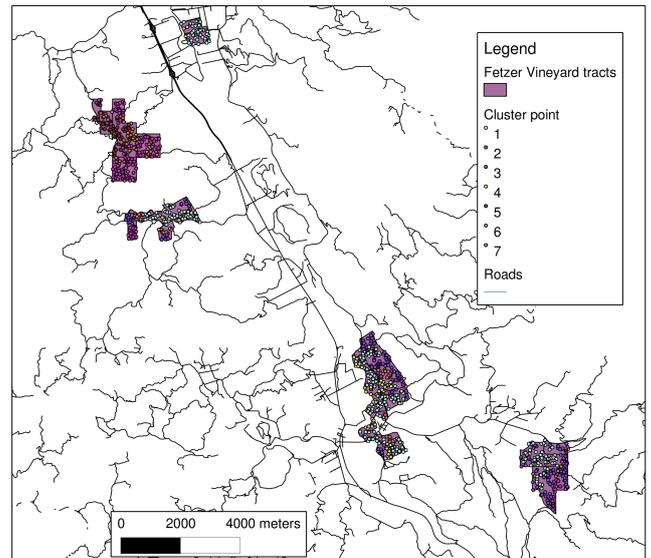


Figure 2: Labeled Cluster Points for Fetzer Study Area

Rather than estimating by eye what cluster point is nearest a given location, it is possible to create a surface from the cluster points and assign a cluster to a location directly. One method for doing this is the Voronoi tessellation technique outlined. This was not done in the case study, but Figure 3 shows the result of doing this for the cluster points in the Fetzer study region.

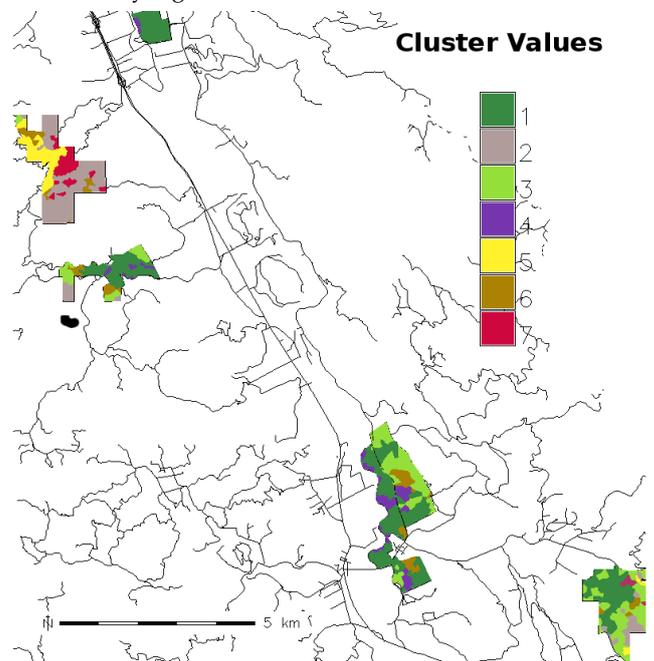


Figure 3: Voronoi Tessellation of Fetzer Cluster Points

Conclusion

This example illustrates how one can use the PAM cluster analysis technique with a set of GIS layers to guide field sampling for research in landscape ecology. We have used this technique to sample other agroecosystems, for example in a study of soil ecology along streams and canals of an agricultural landscape in Yolo County, California (Culman et al.

Number of Vegetation Plots	Number of Soil Pits	Habitat Type	Elevation (meters)	Slope (degrees)	Solar Radiation ($w\ hr/m^2/yr$)
0	19	Vineyard I	173	0.7	2.18×10^6
0	6	Vineyard II	572	10.7	1.99×10^6
0	2	Annual Grassland	213	5.0	2.27×10^6
17	1	Valley Riparian	152	0.4	2.18×10^6
39	2	Mixed Hardwoods I	534	21.3	1.97×10^6
26	13	Mixed Hardwoods II	225	10.7	2.16×10^6
11	1	Mixed Conifer-Hardwood	407	25.8	2.06×10^6

Table 1: Values of Environmental Variables for Cluster Medoids in Fetzer Study

(2010), Young-Mathews, Culman, Sánchez-Moreno, O'Geen, Ferris, Hollander, and Jackson (2010)), and in ongoing work examining soil ecology and gaseous exchange in vineyards in Napa and San Joaquin counties in California. Many more sophisticated variations on cluster analysis could be used: for example, there are a number of algorithms that use spatial constraints directly as part of the clustering process (e.g. Guo and Wang (2011)). But as the example shows, in many cases the results of the clustering are only used as a rough guide for later work, and adopting a simple algorithm is advantageous. An aim for later work will be to automate this procedure. This is made possible by using the GRASS Python scripting API to facilitate the interaction with R in performing the PAM analyses.

Acknowledgements

I would like to thank Louise E. Jackson, John N. Williams, Steven W. Culman, and Kerri Steenwerth for the discussions and problem definition that led to the development of the PAM regionalization algorithm. Support for the Fetzer study was provided by Brown-Forman Corporation, Fetzer Vineyards, and the Kearney Foundation of Soil Science with additional support from the USDA-ARS and the University of California Agricultural Experiment Station.

Bibliography

- S. Culman, A. Young-Mathews, A. Hollander, H. Ferris, S. Sánchez-Moreno, A. O'Geen, and L. Jackson. Biodiversity is associated with indicators of soil ecosystem functions over a landscape gradient of agricultural intensification. *Landscape Ecology*, 25(9):1333–1348, November 2010. ISSN 0921-2973. URL <http://dx.doi.org/10.1007/s10980-010-9511-0>.
- J. C. Duque, R. Ramos, and J. Suriñach. Supervised regionalization methods: A survey. *International Regional Science Review*, 30(3):195–220, July 2007. ISSN 0160-0176. doi: 10.1177/0160017607301605. URL <http://irx.sagepub.com/cgi/doi/10.1177/0160017607301605>.
- J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, December 1971. ISSN 0006-341X. doi: 10.2307/2528823. URL <http://www.jstor.org/stable/2528823>.
- D. Guo and H. Wang. Automatic region building for spatial analysis. *Transactions in GIS*, 15:29–45, July 2011. ISSN 13611682. doi: 10.1111/j.1467-9671.2011.01269.x. URL <http://doi.wiley.com/10.1111/j.1467-9671.2011.01269.x>.
- R. J. Johnston. Grouping and regionalizing: some methodological and technical observations. *Economic Geography*, 46(supplement): 293–305, 1970.
- L. Kaufman and P. Rousseeuw. *Finding groups in data an introduction to cluster analysis*. Wiley, Hoboken, N.J., 2005. ISBN 9780471735786.
- M. A. Patten and B. D. Smith-Patten. Biogeographical boundaries and Monmonier's algorithm: a case study in the northern Neotropics. *Journal of Biogeography*, 35(3):407–416, March 2008. ISSN 0305-0270. doi: 10.1111/j.1365-2699.2007.01831.x. URL <http://doi.wiley.com/10.1111/j.1365-2699.2007.01831.x>.
- Ş Procheş. The world's biogeographical regions: cluster analyses based on bat distributions. *Journal of Biogeography*, 32(4):607–614, March 2005. ISSN 03050270. doi: 10.1111/j.1365-2699.2004.01186.x. URL <http://doi.wiley.com/10.1111/j.1365-2699.2004.01186.x>.
- Climate Action Reserve. Forest project protocol, version 3.1. Technical report, Climate Action Reserve, Sacramento, CA, 2009.
- P Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, November 1987. ISSN 03770427. doi: 10.1016/0377-0427(87)90125-7. URL <http://linkinghub.elsevier.com/retrieve/pii/0377042787901257>.
- N. A. Spence and P. J. Taylor. Quantitative methods in regional taxonomy. *Progress in Geography*, 2:1–63, 1970.
- D. E. Stooksbury and P. J. Michaels. Cluster analysis of southeastern U.S. climate stations. *Theoretical and Applied Climatology*, 44(3-4): 143–150, 1991. ISSN 0177-798X. doi: 10.1007/BF00868169. URL <http://www.springerlink.com/index/10.1007/BF00868169>.
- GRASS Development Team. Geographic resources analysis support system (GRASS) software., 2009a. URL <http://grass.osgeo.org/>.
- R Development Core Team. R: A language and environment for statistical computing, 2009b. URL <http://www.r-project.org/>.
- A. Trakhtenbrot and R. Kadmon. Environmental cluster analysis as a tool for selecting complementary networks of conservation sites. *Ecological Applications*, 15(1):335–345, February 2005. ISSN 1051-0761. doi: 10.1890/04-0077. URL <http://www.esajournals.org/doi/abs/10.1890/04-0077>.
- G. C. van Kooten. Biological carbon sequestration and carbon trading re-visited. *Climatic Change*, 95(3-4):449–463, April 2009. ISSN 0165-0009. doi: 10.1007/s10584-009-9572-8. URL <http://www.springerlink.com/content/8372834132127434/>.
- R.I. Vane-Wright, C.J. Humphries, and P.H. Williams. What to protect? — systematics and the agony of choice. *Biological Conservation*, 55(3):235–254, January 1991. ISSN 00063207. doi: 10.1016/0006-3207(91)90030-D. URL <http://linkinghub.elsevier.com/retrieve/pii/000632079190030D>.
- J. N. Williams, A. D. Hollander, A. T. O'Geen, L. A. Thrupp, R. Hanifin, K. Steenwerth, G. McGourty, and L. E. Jackson. Assessment of carbon in woody plants and soil across a vineyard-woodland landscape. *Carbon Balance and Management*, 6(11), 2011.
- S.E. Wiltshire. Identification of homogeneous regions for flood frequency analysis. *Journal of Hydrology*, 84(3-4):287–302, May 1986. ISSN 00221694. doi: 10.1016/0022-1694(86)90128-9. URL <http://>

linkinghub.elsevier.com/retrieve/pii/0022169486901289.
A. Young-Mathews, S. W. Culman, S. Sánchez-Moreno, A. O'Geen,
H. Ferris, A. D. Hollander, and L. E. Jackson. Plant-soil bio-
diversity relationships and nutrient retention in agricultural ri-
parian zones of the sacramento valley, california. *Agroforestry
Systems*, 80(1):41–60, July 2010. ISSN 0167-4366. doi: 10.
1007/s10457-010-9332-9. URL <http://www.springerlink.com/>

content/m57j684058401536/.

Allan D. Hollander
Information Center for the Environment
University of California, Davis
adhollander@ucdavis.edu

Functional Coverages

Gennadii Donchyts, Fedor Baart, H.R.A (Bert) Jagers, Arthur van Dam

Abstract

A new Application Programming Interface (API) is presented which simplifies working with geospatial coverages as well as many other data structures of a multi-dimensional nature. The main idea extends the Common Data Model (CDM) developed at the University Corporation for Atmospheric Research (UCAR). The proposed function object model uses the mathematical definition of a vector-valued function. A geospatial coverage will be expressed as a vector-valued function whose dependent variables (the vector components) are fully defined by its independent variables (the coordinates).

Our goal is to provide an API using a terminology and an object model that is both appealing to computer scientists and numerical modelers and is flexible enough to enable defining data structures for a wide range of applications. Examples of such data structures can be: wind velocity as a continuous variable defined along the channels in a river network. Precipitation data defined as a time-dependent variable on a set of sub-catchments of a drainage basin, preserving association with sub-catchment features.

The new object model provides a basis for both continuous and discrete coverages including non-geospatial data structures such as time series. Different storage models for variables are implemented, based on the Network Common Data Format (NetCDF), the Geospatial Data Abstraction Library (GDAL) and memory.

The API is available as set of open source libraries developed in C# consisting of a multi-dimensional arrays library; a scientific data structures library defining variables, functions, units of measure; a geospatial extensions library built on top of GeoAPI.NET and NetTopologySuite, defining specialized coverages: network coverage, feature coverage, regular grid coverage, and unstructured grid coverage.

1 Introduction

Geospatial coverage is the concept describing geographic phenomena upon which a range of data values can be present. The concept dates back to early ArcInfo³⁵ versions, but has been applied and rethought numerous times in data standards and Application Programming Interface (API)s ever since. The assumptions that were made in these do not always make them easily applicable in new fields, where both the data and the spatial domains may be more complicated. One special example is the role of the time dimension, which was absent in early coverage definitions. Over the past decades, many initiatives were taken to incorporate time into geospatial applications ((Peuquet, 1999), (Wachowicz, 1999), (Goodall et al., 2004)) resulting in a sometimes overly special role of the time dimension.

The goal of the present work is to define a high-level API that will allow presenting most of the existing geospatial

and temporal coverage data types in terms of vector-valued functions of one or more independent variable(s), inspired by basic vector calculus. The main reason for doing this is a unification of the storage of spatiotemporal domains and coverage values on top of these domains. In addition this will simplify development of generic data transformation algorithms such as aggregations, filtering, interpolation, and extrapolation. It will be much easier to re-use them, which will simplify development of applications, as was already shown for example for the Open Modeling Interface (OpenMI) 2.0 by Donchyts et al. (2010). We will show that this higher-level API can not only define scalar data on spatial networks or grids, but can also be used to define more general data structures, such as time-series or vector fields. Evidently, these can be used in geospatial applications, but are also applicable in a wider scope.

Related work

The definition and use of a geospatial coverage can be approached from several angles; two existing standards are the ISO 19123 standard (schema for coverage geometry and functions) and the Common Data Model (CDM), designed by the University Corporation for Atmospheric Research (UCAR).

The ISO 19123 standard defines a coverage as a "coverage is a feature that associates positions within a bounded space (its domain) to feature attribute values (its range)". ISO 19123 as well as the Open Geospatial Consortium (OGC) Abstract Specifications (2007) define an abstract concept of the coverage object model and how it supports mapping from a spatial, temporal or spatio-temporal domain to feature attribute values. These standards are mainly used to form the basis of an geospatial API's such as GeoAPI.³⁶

Alternatively, the CDM API mainly focuses on the multi-dimensional aspects of coverages. An overview of the UCAR CDM and its mapping into the corresponding elements of the international standard coverage data model of ISO 19123 is presented by Nativi et al. (2008). More information about CDM and the Network Common Data Format (NetCDF) can be found in Rew and Davis (1990).

As a result of these two different angles, the coverage API is very well suited in geospatial applications, whereas the CDM API has proven itself in numerical applications and for large datasets, see, e.g., Rutledge et al. (2006) and Signell et al. (2008), respectively. For applications where data structures are less organized (ecological models) or require more complex relations (river models), both data models are less applicable. This paper deals with datasets where information is dependent on time and covers a network- or grid-based area (the spatial domain). Examples are: ocean dynamics (water levels and velocities in 3D grid layers) and river flows (water levels, velocities and transport across network connections), see, e.g., Kernkamp et al. (2011). The new API should facilitate this.

An example of an often-used API for time-dependent data sets defined on complex geometries can be found in the open source Visualization Toolkit (VTK) (Schroeder et al., 2000).

³⁵<http://www.esri.com/software/arcgis/arcinfo>

³⁶<http://www.geoapi.org/>

Each object in VTK maintains an internal time stamp that is automatically updated when the object state changes (usually as a result of setting an instance variable value). An alternative approach to dealing with the time dependent grid based data is presented by [Howe and Maier \(2005\)](#). They use a relational database based approach using an algebra notation to manipulate both regular and irregular gridded datasets. When comparing these [API](#)'s and data models we can see that they differ in scope, detail and field. When approached from the visualization field as was done in [VTK](#), the time dependency and geographic aspects were added later (see for example `vtkGeoGlobeSource`, `vtkTemporalDataSet`). Because the [API](#) is focused on visualizations we can see that all geometry objects are well defined and easy to comprehend.

When approached from the Geographic Information System (GIS) field we can see that the Feature is the base of all relevant objects, however it is still ongoing discussion about what it is and how it should be implemented and used in applications. [OGC Coverage Primer \(Nordgren, 2006\)](#) reflects this in the following way: *The question "What is a feature?" leads directly to a philosophical rabbit hole which deposits the unwary questioner in a wonderland from which it is difficult to return.* Also, time aspects are still being introduced in the [OGC](#) standards and as a result most widely used geospatial [API](#)'s that build upon these [OGC](#) Standards still lack support for time dependency.

It is important to note that computer scientists will often define their own [API](#) and naming conventions that are (partly) based on the problem domain in which they happen to be working at the time rather than using the terminology of the generic underlying concepts. This complicates the reuse by people in other domains. The idea to use multi-valued functions to represent data structures in a more generic way has been formulated by [Treinish \(1999\)](#) as: "Any data set may be considered as a single or multi-valued function of one or more independent variable(s)". In the present work we try to generalize and expand this idea to be applicable to geospatial coverages, resulting in both a conceptual description and an implementation in the form of a class library which can be easily re-used in geospatial applications.

Outline

Section 2 motivates the approach we took in defining our [API](#) and summarized the underlying ideas. Sections 3 and 4 then respectively describe the generic vector-Function [API](#) and the geospatial [API](#) on top of it. Section 5 describes how the functions and their values can be stored in memory and in files. Section 6 considers our resulting [API](#) and provides some additional motivation. Section 7 summarizes the paper in several conclusions. The appendices summarize the terminology and acronyms used throughout the paper.

2 Method

The most important question is how to define an [API](#) that offers data types and functionality for generic mathematical concepts such as variables, and at the same time be specifically suitable as well for both geospatial and non-geospatial applications? Can we introduce time-dependency in a non-intrusive way? The idea is to define a common [API](#) which will describe all generic mathematical data structures required to

manage variable values and then base a geospatial coverage [API](#) on top of that to provide a better separation of concepts and better code re-use.

The spatial domains also pose challenges: what to do when the topology is complicated, for example a river network? The channels in this network form a set of interconnected features, each of which is a polyline or polygon geometry as shown in [Figure 1](#). Can we still use the same classes to model this situation in a similar way as we would do it for a simple time series at a single point station?

Next comes the data defined on the domains. In our river example, consider a time-dependent wind velocity field defined as a continuous vector variable $\vec{V} = (v_x, v_y)$ (discrete values with interpolation) along the channels in the river network. Inter- and extrapolation on such complicated geometries is also nontrivial. Do we need to deal with spatial variables differently than with time variables?

Even if it would be clear what should be done in this example, it is still not a trivial task to define using an object-oriented language. Both the [OGC Coverage model](#) and [CDM](#) fail to define it completely. The [OGC Coverage model](#) is not flexible enough in the sense that it introduces many classes but there is no conceptual basis where every type of Coverage would fit. In terms of the [CDM](#), it should be trivial to define values of all variables used here (river coordinates and wind) using multi-dimensional arrays. However, there is no room for the rest of the meta-information (the river topology and offsets of network locations). The use of basic [CDM](#) attributes is insufficient here.

We will show that using the new approach proposed in this paper all coverage types can be defined in terms of vector-valued functions.

Design steps

Before introducing an [API](#) for multi-dimensional data structures, we will first try to identify and analyze the actual problem domains related to the practical applications that involve coverages or multi-dimensional data structures ([Figure 2](#)). Then we will try to identify the functionality required by the developers when developing applications related to these domains.

For the design of the [API](#) we used the following steps: identification of entities / classes, construction of a reference implementation of the classes to match the different fields (geospatial, environmental), separation of core logic of the classes from persistency so that multiple storage choices can be used, identification of interfaces / classes which should belong to the [API](#).

The term Coverage is used in the geospatial domain (by the [OGC](#)) to describe discrete or continuous characteristics of the real world features. We will try to match our [API](#) as much as possible (on a conceptual level) to the requirements listed in the [OGC](#) standards.

Before moving to the geospatial domain we will first try to analyze in details how vector-valued function can be defined in terms of the software component since we plan to use it as a basis of our [API](#). After that we will show that the new [API](#) is very well suitable to describe any coverage used in the geospatial domain.

When developing an [API](#) for multi-dimensional structure an important aspect is persistency. In section 5 [Persistence](#)

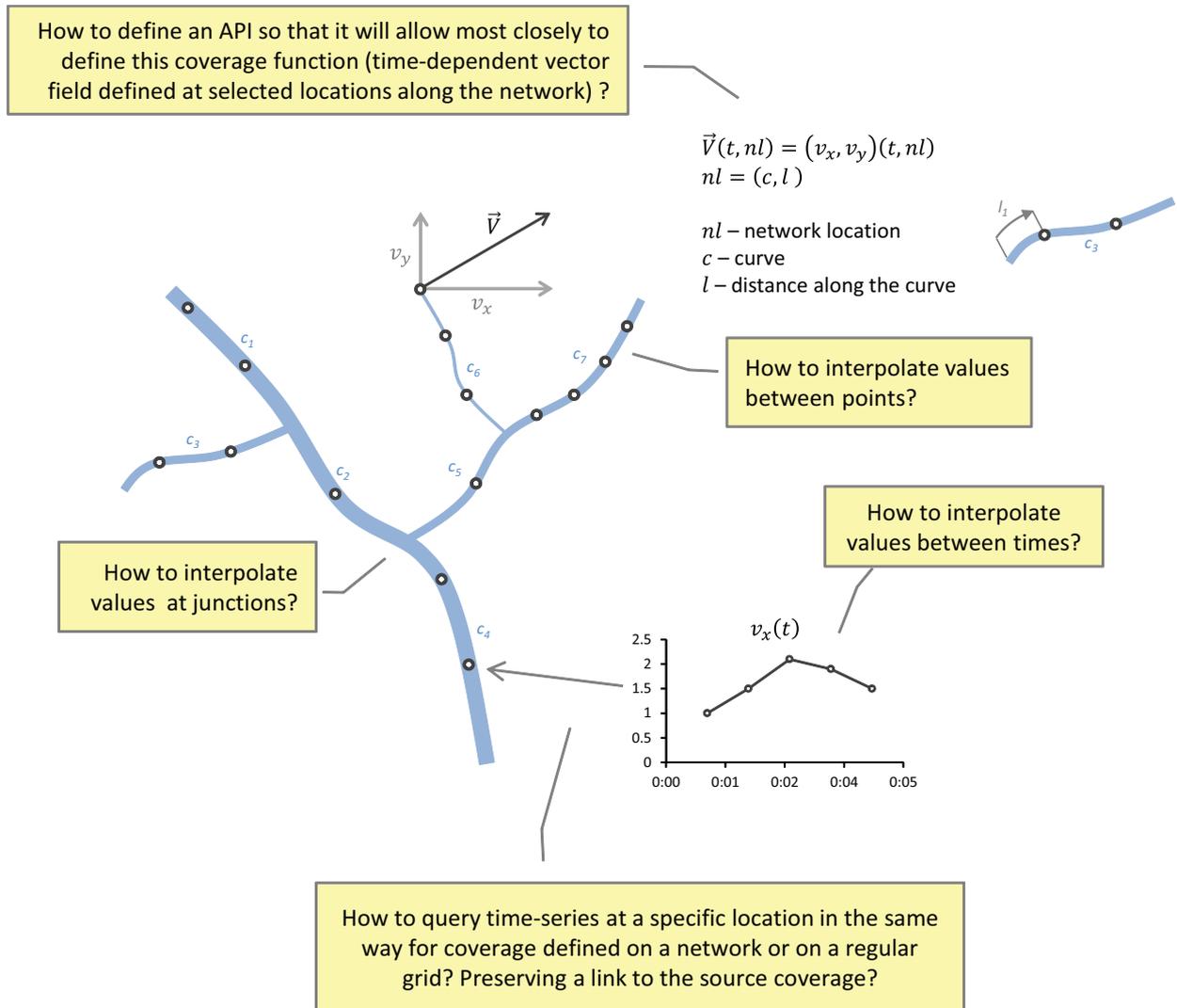


Figure 1: Problems appearing while modeling variables defined on the network

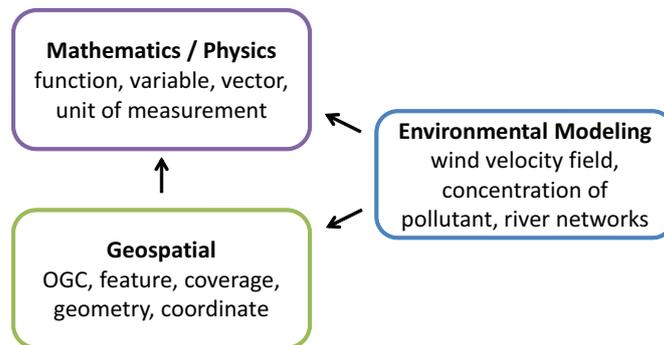


Figure 2: Problem domains involved

several storage options will be discussed allowing reuse of existing data access libraries or file formats to persist structure and values of the vector-valued functions.

We will use motivating example problems appearing in the environmental modeling field, see sections 3.2 and 4.

3 Vector-valued Function

3.1 Mathematical Definition

The main idea of the new data structure is based on the following statement:

Any data structure of a multi-dimensional nature can be presented as a single or multiple number of a vector-valued

functions.

In vector calculus a vector-valued function is defined in general form as Equation 1

$$F = (f_1, f_2, \dots, f_n)(x_1, x_2, \dots, x_m) \quad (1)$$

where x_1, x_2, \dots, x_m are independent variables and f_1, f_2, \dots, f_n are dependent variables.

Typical examples of independent variables are time, spatial coordinates, categories, etc. Dependent variables usually represent actual quantities. In scientific applications these variables are very frequently sampled or discretized and thus their values are defined in a form of arrays together with information about how to interpolate values of dependent variables where no exact values of the independent variables are available. For arguments (independent variables) these arrays are in most cases one-dimensional ($R1$) while for components (dependent variables) dimensionality of arrays is defined by a number of arguments used (Rm). In many cases value type used by the variable is a real number (R) however in general it can be any type available in programming language, for example: string, class, feature, etc. In some cases the rank of the independent variable space may be different from the rank of the sampled independent variable space: discrete coordinate arrays may be rank 1 (regular grid), or rank 2 (curvilinear/irregular grid), or even rank 3 (time dependent moving grid).

Let us analyze vector-valued function in more details: consider that we want to define a time-dependent velocity field (e.g. 3 moments in time), defined on a discrete, regular grid (see Figure 3). In order to store values of this function we will have first to decompose it into independent (x , y and t) and dependent (v_x and v_y) variables. We can see that independent variables have to be defined as a set of values (ordered set if we want to interpolate values of the dependent variables along the argument). On the other hand dependent variables require 3-dimensional arrays to store their values (number of independent variables used in function).

It is simple to show that every variable can be also considered as a function, as result we can list all objects required to store the above example, as can be seen in Table 1.

In the table Arguments denote independent variables and components - dependent variables.

In general any variable used in a function (independent or dependent variable) can be described by a set of properties: value type, units of measure, typical minimum and maximum values.

Additionally, for every independent variable we need to define interpolation and extrapolation method since its values are defined only at discrete locations. This will allow computing values of dependent variables outside of independent variable values space.

From the table we can see that in order to completely define all objects used in this example we will have to define all functions listed in the table, together with their properties as well as relations between them such as that some of them are used as independent variables and other as dependent variables. In case if we have a function that uses more than one component (V) - the only thing to be stored is its relation to child component variables since its values are completely defined by the values of its components (v_x, v_y)

3.2 API

Based on the points discussed in the previous section we believe that the class diagram presented in Figure 4 most closely describes all objects required to introduce a vector-valued function.

As can be seen from Equation 1, component variables can be actually seen as vector-valued functions by themselves.

The code listing in Figure 5 shows how the API can be used. This is a simple example that shows how we can define a variable, its properties and an array of values.

Since a Variable in our API is automatically considered to be a Function - we can also start combining variables as shown in Figure 6.

This type of function is one of the most frequently used. A simple example from hydrodynamic modeling can be a water level defined as a function of time, e.g. measured at some location: $y=y(t)$. In this case value type of the argument variable will be DateTime (C#) instead of double. In this case a water level is dependent variable and time is independent variable.

If we make it a bit more complex we can measure water level on a moving boat. In this case a water level variable is measured as a function of time, but so is location. So we have location as an object and a water level as a scalar value defined as a function of time: $F=(location, depth)(t)$.

Suppose we measure wind direction at a meteorological station, then we have two parameters for direction and speed or in Cartesian space a u and v part of the vector. Both again are defined as functions of time.

The source code required to work with a vector-function that uses more than one independent variable (components) does not look much different, see Figure 7.

Note that API provides different ways to access or assign values of the variables. Depending on performance requirements values of the variable can be set as an array at once or one by one using simple and intuitive syntax.

4 Adding geospatial aspects to Function

Even though the above API is powerful enough to describe a vector-valued function, in some cases we need to extend it in order to apply it to other domains. We will try to define a Coverage types on top of the Function API.

4.1 What is Coverage?

Definition of Coverage in many geospatial applications is also confusing and is given on a very conceptual level. We will introduce term Coverage as a bridge between two worlds: Geospatial and Mathematical. The UML class diagram is shown in Figure 8. The nice thing is that if Coverage can be defined as a function - then its values can be accessed in the same way as in examples of the previous section. On the other side Coverage is a geospatial object, which means it has to extend functional part with geometries. For example Geometry property of the coverage can be either a geometry that defines bounds of the coverage, or a complex geometry representing every location where values of the coverage are defined (GeometryCollection).

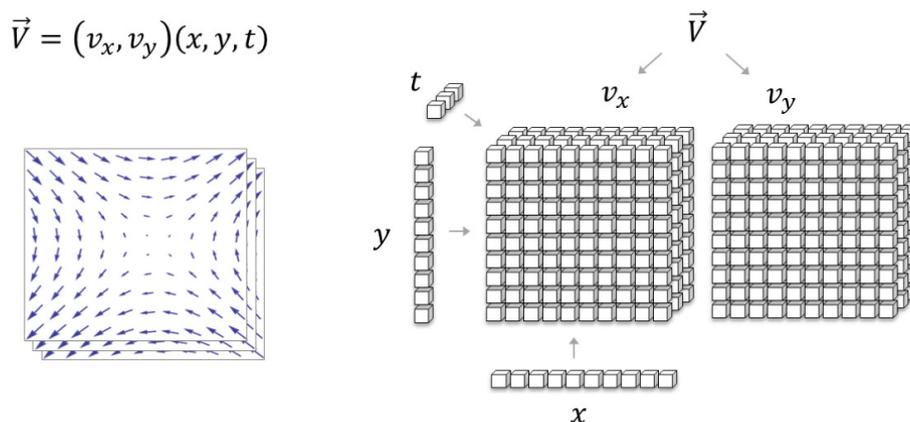


Figure 3: Analysis of the data structures required to store discrete vector-valued function

Table 1: List of functions and their properties used in example

Function	Definition	Arguments	Components	Rank	Unit
x	(x)	0	1	1	L
y	(y)	0	1	1	L
t	(t)	0	1	1	T
v_x	$(v_x)(x, y, t)$	3	1	3	L/T
v_y	$(v_y)(x, y, t)$	3	1	3	L/T
\vec{V}	$(v_x, v_y)(x, y, t)$	3	2	3	L/T

In the OGC standards there is a clear distinction made between continuous and discrete coverages. We do not see the need to separate them. Really, the difference between if coverage is discrete or continuous is just an interpolation type used for its arguments. In fact we can define a Coverage that is discrete along one argument and continuous (interpolated) along the other one.

All other coverage types can be very simply defined as an extension of the ICoverage interface. For the sake of space we will not list all UML diagrams here.

4.2 Regular Grid Coverage

In many applications, including numerical modeling, data of the models are often defined on a discrete grid. Suppose we compute water level which can change in time and is defined on a rectangular regular grid, then it can be defined as a function of x, y coordinates and time t . In these case x and y identify location and used as independent variables as shown in Figure 9. Actually this is also true for rectilinear grid, where values of x and/or y are not equidistant, see (Balaji and Liang, 2006). In case of regular grid the values of x and y variables (arguments) are equidistant and as a result their storage can be simplified.

Depending on a type of grid we can also use cells of the grid (objects) as an independent variable values instead of scalar x and y variables to identify location on a grid. See 4.5 section for an example.

The biggest advantage of using the same base API to work with Coverages is that the functionality of Coverage can be very easily extended. For example in order to make regular grid coverage time dependent we only need to add an additional argument (independent variable) of a time value type, see Figure 10. The rest remains the same.

Of course in case if we have other functionality (for example rendering) based on a specific Coverage type - we will need to extend it a little to make sure that we accessing only with a values corresponding to a single time value.

4.3 Feature Coverage

We will call FeatureCoverage a function where one of arguments uses a Feature as a value type.

Consider the following class as an example: a City, which has some default properties such as Name and Population (these properties are also available as feature attributes and accessible via Attributes dictionary).

Now imagine that we want to compute a total precipitation over city for a given period of time without modifying existing City features. It can't be simply added as an attribute (a property) since it doesn't seem to be a default characteristic of the City. In this case FeatureCoverage type can be used to define a coverage function that uses cities as values of an independent variable as show in Figure 11 and Figure 12.

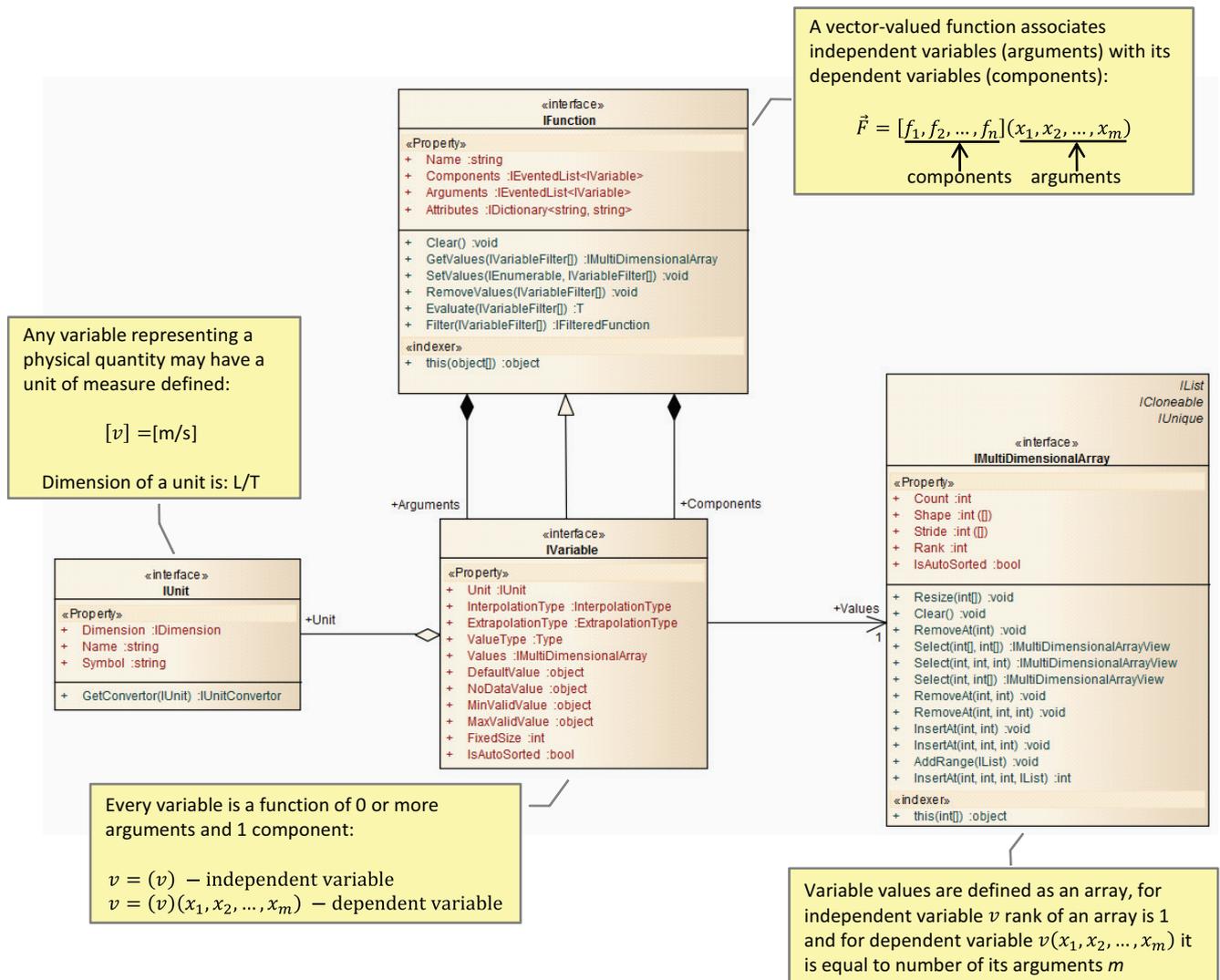


Figure 4: Vector-value function class diagram

4.4 Network Coverage

In some cases variable values are defined on a set of curves (see OGC CurveCoverage). In more specific cases these curves are connected into a network or graph, for example: river networks, roads and pipelines. It is frequently necessary to define a Coverage that can be seen as a continuous function defined along the branches (the polylines) of such a network (see Figure 13).

This example is a bit more complicated compared to the ones discussed in the previous sections. The main problem is that we do not have an explicit independent variable to depend on in our function. Still we can introduce a variable which is defined as a combination of a curve and offset along that curve. In this case it will uniquely define location on a network.

The only remaining problem is that if we would like to evaluate values of the dependent variables in other but existing locations on a network - we will need some custom interpolation algorithm at the sections where branches connect with each other. Once it is implemented - the rest of the functionality works as in any other function type.

Additionally to the network location independent variable we can also add time or any other variable.

We've defined NetworkCoverage as a separate coverage type as shown in Figure 14. However it will be more correct to introduce a CurveCoverage type first, and then define NetworkCoverage as an extension to it, in case if we need to use some network-specific interpolations e.g. at the nodes connecting different branches of the network. In this case it will be more consistent with the ideas introduced in the OGC Coverage standards.

4.5 Unstructured Grid Coverage

We did not fully implement support for unstructured grid coverages yet using new API, but it should not be any more complicated than previously shown examples. In fact UnstructuredGridCoverage can be implemented in a way similar to FeatureCoverage, which depends on grid cells or interfaces between cells, depending on where values are defined (see Figure 15). Additionally, custom interpolation methods have to be implemented for e.g. IDW, Krigging or any other interpolation methods required to evaluate values outside of the

```

[Test]
[Category("Example")]
public void SimpleVariable()
{
    var x = new Variable<double>
    {
        Name = "x",
        Unit = new Unit("meter", "m"),
        Values = { 1.0, 2.0, 3.0 }
    };

    // access values array used by variable x
    IMultiDimensionalArray<double> values = x.Values;

    // asserts
    values[1]
        .Should("2nd value").Be.EqualTo(2.0);

    values.Rank
        .Should("rank of a values array is 1").Be.EqualTo(1);

    values.Count
        .Should("number of values").Be.EqualTo(3);
}

```

$x = \{1.0, 2.0, 3.0\}$
unit of measure of x is *meter*

Figure 5: Create and set values of a simple scalar variable (C#)

```

[Test]
[Category("Example")]
public void DependentVariable()
{
    var x = new Variable<double> { InterpolationType = InterpolationType.Linear };
    var y = new Variable<double> { Arguments = { x } };

    // set values using value index operator
    y[1.0] = 100.0;
    y[2.0] = 200.0;
    y[3.0] = 100.0;

    // set values using method (replaces existing values of y)
    y.SetValues(new [] { 100.0, 200.0, 300.0 });

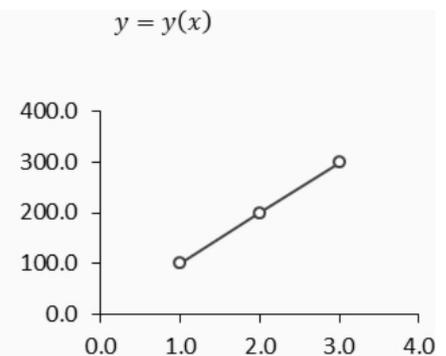
    // access values using value index operator
    y[1.0]
        .Should("exact value y(x = 1.0) = 100.0").Be.EqualTo(100.0);

    y[1.5]
        .Should("interpolated value y(x = 1.5) = 150.0").Be.EqualTo(150.0);

    // access values of y as an array
    x.Values
        .Should("check x values").Have.SameSequenceAs(new[] { 1.0, 2.0, 3.0 });

    y.Values
        .Should("check y values").Have.SameSequenceAs(new[] { 100.0, 200.0, 300.0 });
}

```

Figure 6: Create a 1d function $y=y(x)$, use linear interpolation for x (C#)

argument values domain.

5 Persistence

In order to store values of the functions an interface `IFunctionStore` was introduced as a part of [API](#), see [Figure 16](#). An implementation of the `IFunction` / `IVariable` uses `IFunctionStore` to access all functions available in the store as well as to

access their values.

Currently the following implementations are supported: `MemoryFunctionStore`, `NetCDFFunctionStore` and `GdalFunctionStore`. The first implementation is a default one and simply keeps a set of multi-dimensional arrays as well as a set of function objects in memory. The second is used to store functions in the [NetCDF](#) files, wrapping UCAR Java implementation converted to .NET on a byte-code level using `IKVM.NET`.

```

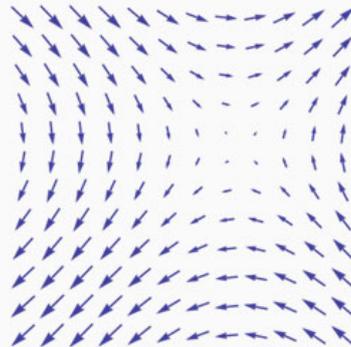
[Test]
[Category("Example")]
public void VelocityField2D()
{
    var x = new Variable<double> { Values = { 1.0, 2.0 } };
    var y = new Variable<double> { Values = { 1.0, 2.0, 3.0 } };
    var vx = new Variable<double>();
    var vy = new Variable<double>();

    var velocity = new Function
    {
        Components = { vx, vy },
        Arguments = { x, y }
    };

    // set vx, vy values as an array
    vx.SetValues(new[] { 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 });
    vy.SetValues(new[] { 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 });

    // set vx values using indexer
    vx[1.0, 1.0] = 100.0; vx[2.0, 1.0] = 400.0;
    vx[1.0, 2.0] = 200.0; vx[2.0, 2.0] = 500.0;
    vx[1.0, 3.0] = 300.0; vx[2.0, 3.0] = 600.0;
}
    
```

$$\vec{V} = (v_x, v_y)(x, y)$$



Arrays

v_x	□□
v_y	□□
x	□
y	□

Figure 7: Define 2d velocity field function (C#)

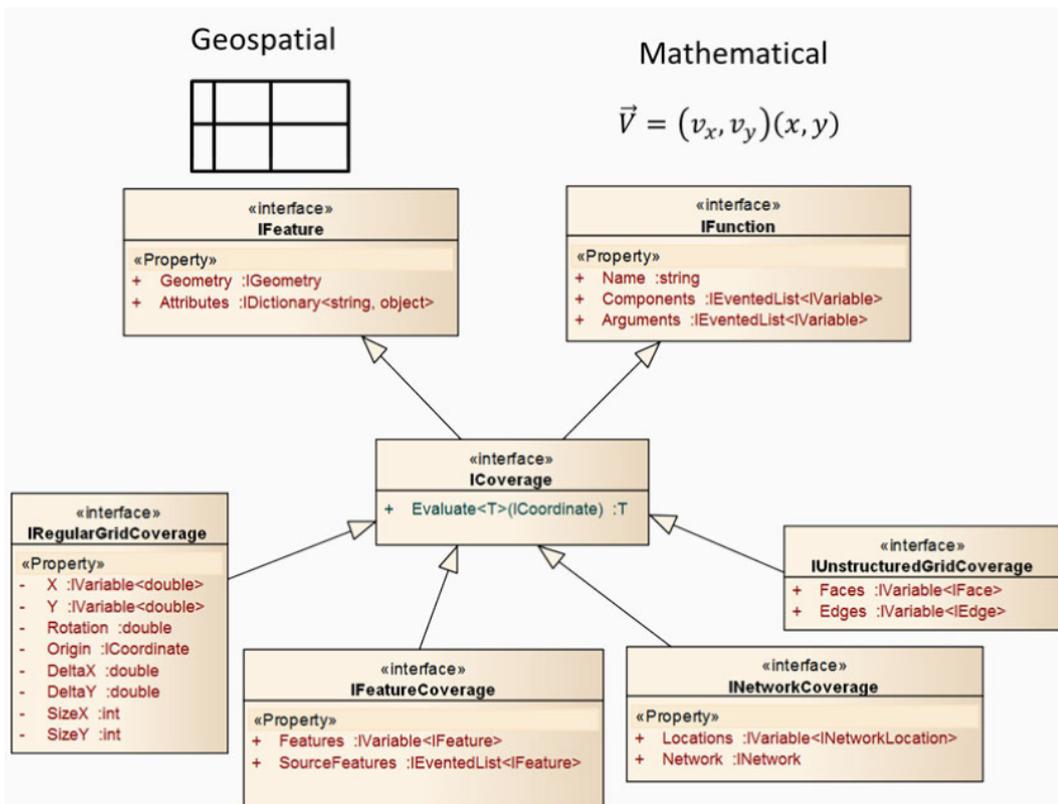


Figure 8: Coverage Class Diagram

The third implementation is used to access raster data stored in GDAL file formats.

5.1 Store Functions in NetCDF

Because the Common Data Model is a concept behind the NetCDF file format, the file format is almost a perfect match for as a storage for Function classes. Still the following information needs to be defined implicitly using attributes:

Relations between container vector-valued function and its component variables.

Relations between components and arguments. Even it can be reconstructed using NetCDF dimensions; it is still error prone and not very intuitive.

Custom type mappings, in case if we want store entities (objects) in the NetCDF variables.

In the last case NetCDF attributes can be used using some

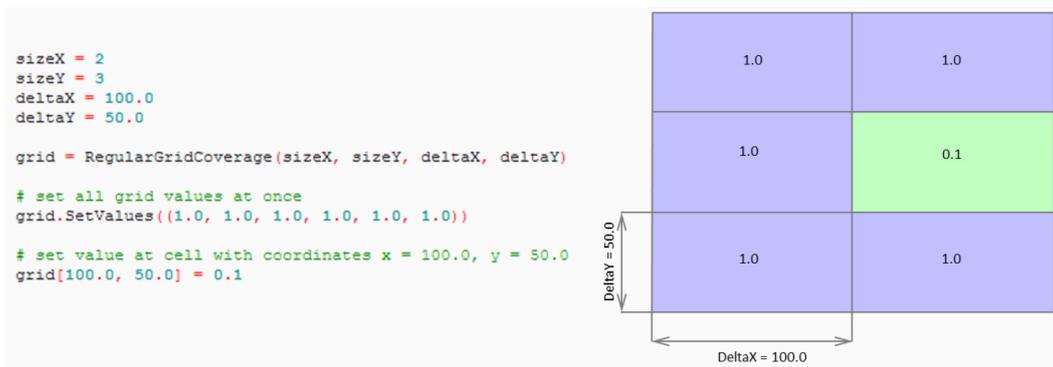


Figure 9: RegularGridCoverage defined as 2D function (Python)

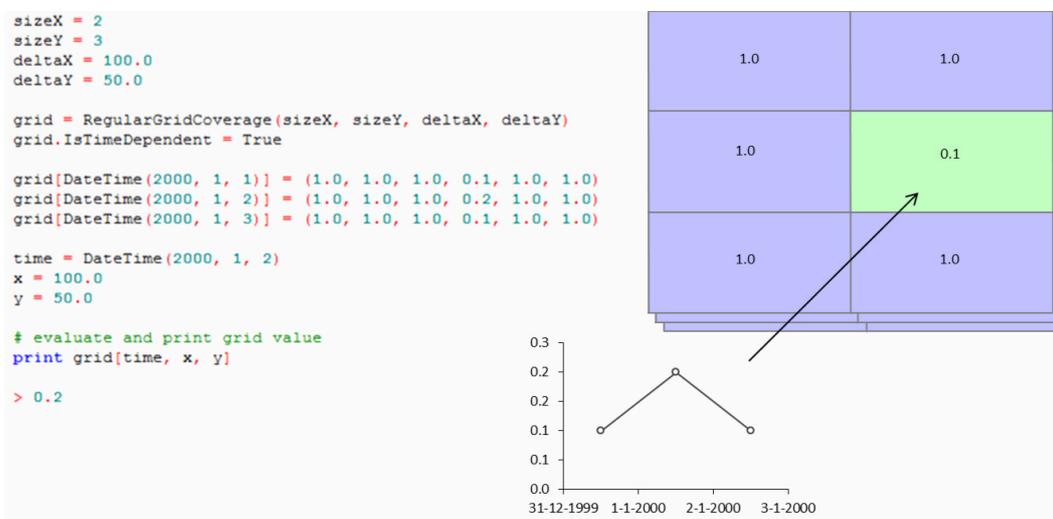


Figure 10: Define and query value of a time-dependent RegularGridCoverage (Python)

convention in order to define where properties need to be stored and to which type they belong to.

6 Discussion

The need to this API was mainly dictated by the reality that there was no API library available written in C# which is comparable to NetCDF. Instead of reinventing the wheel we tried to extend concepts used in the CDM. On the other hand we did not port (or wrap) existing NetCDF API in C# but tried to make it more intuitive by reviewing the concepts behind it. The NetCDF library is a very powerful one but we believe that the API introduced here better represents the reality (or mathematical abstractions used to describe the reality within computerized applications). The major difference between CDM and the present API, except of course the language and syntax, is that CDM uses Dimensions as a separate entity next to Variable. For the API we present the Dimension is not required as a separate entity. The dimensions can be derived from the vector-valued functions.

Some variables need to be defined not as a set of values but e.g. as an equidistant series with start, stop and step. In this case only several properties need to be stored instead of array. The implementation can still generate all possible values on-the-fly in order to use this specific variable type in

the same way as other variables.

In some cases a single vector-valued function is not sufficient to define all data structures. For example for curvilinear grid we want to preserve the information that the grid cells are defined as a 2D matrix. In this case it will be necessary to combine different vector-valued functions in order to fully define the data structure.

When comparing the proposed API to the OGC/ISO Coverage specifications it is important to note that the latter specifications are quite complex compared to the API presented here. Another point that is missing in OGC Coverages API is that it does not provide a unified way to access all Coverage values in the same way for all Coverage types. We hope that this work will influence the existing OGC Coverage API in a way that it will become simpler to use.

An important aspect that was not discussed in the present paper is related to the definitions of functions which represent filtered version of existing ones. This is a very useful functionality, especially if existing functions need to be queried and e.g. visualized as a function with a smaller number of arguments (time series representing values of a single cell of a time-dependent regular grid). In many cases this filtered functions need to be stored somewhere next to the real functions. Another example is when variable represents an aggregated version of another variable. It becomes a very tricky task when we add a user requirement that the connection between

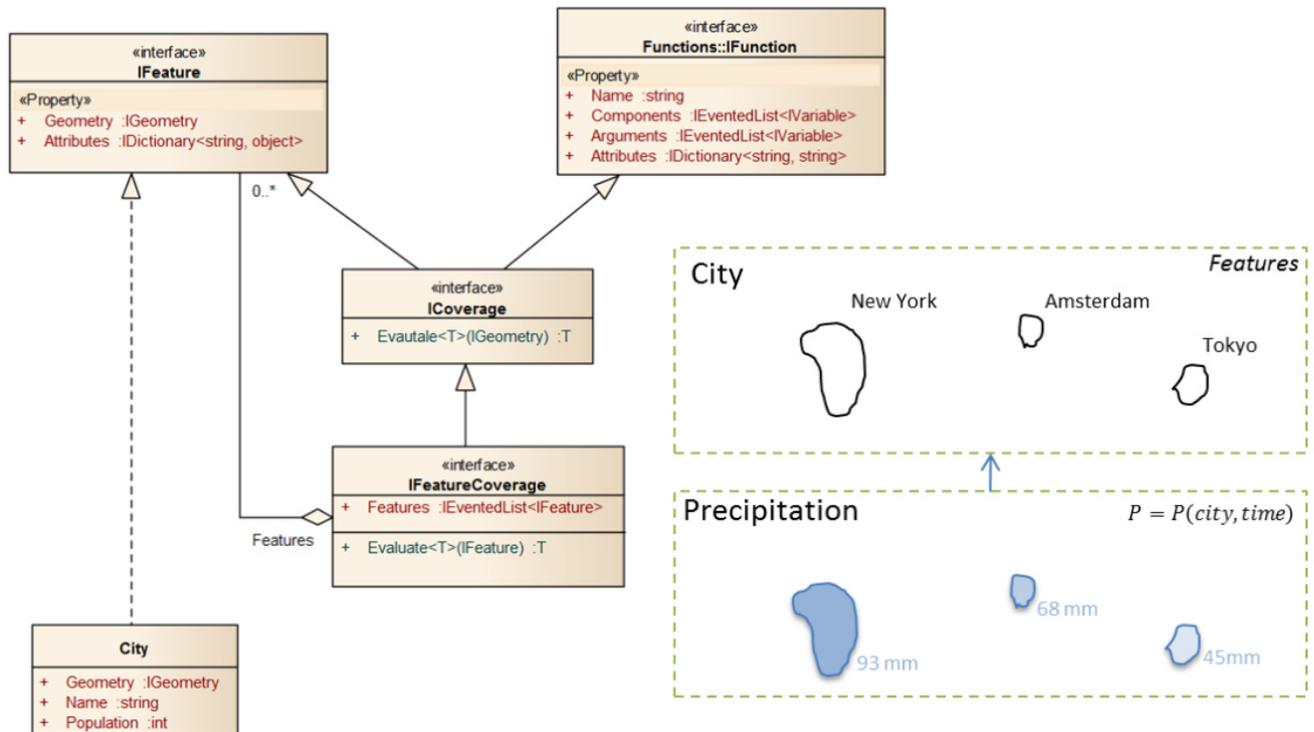


Figure 11: FeatureCoverage, function where one of the arguments uses features as values

```

public class City : Feature
{
    [FeatureAttribute] public string Name { get; set; }
    [FeatureAttribute] public int Population { get; set; }
}

[Test]
[Category("Example")]
public void FeatureCoverage()
{
    var cities = new[]
    {
        new City { Name = "New York", Geometry = new Point(1.0, 1.0) },
        new City { Name = "Amsterdam", Geometry = new Point(2.0, 2.0) },
        new City { Name = "Tokyo", Geometry = new Point(3.0, 3.0) },
    };

    // construct coverage
    var coverage = new FeatureCoverage { Features = cities, Name = "cities" };
    coverage.Arguments.Add(new Variable<City>("city"));
    coverage.Components.Add(new Variable<double>("precipitation"));

    // set values
    coverage[cities[0]] = 45.0;
    coverage[cities[1]] = 68.0;
    coverage[cities[2]] = 93.0;
}
    
```

Figure 12: Creation and use of FeatureCoverage using existing features (C#)

original and filtered variables must be live, meaning that when values in the original variable change - values in the filtered variable will be recalculated automatically.

7 Conclusion

We have presented an API that provides users the possibility to work with geospatial and non-geospatial types of multidimensional data in a convenient way. By providing a direct connection to the NetCDF data format we hope that our API will become especially popular for working with results from numerical models. The library implementation also fills in

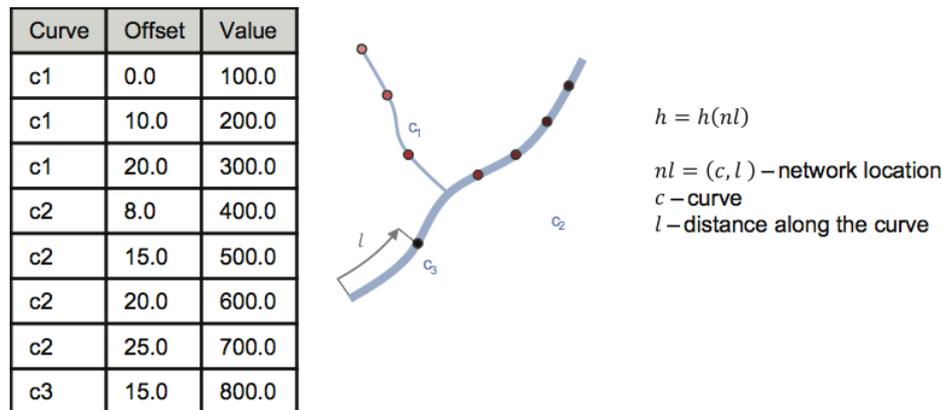


Figure 13: Network coverage, values defined along the network branches.

```

# create simple network containing 4 nodes and 3 branches
network = HydroNetwork(Name = "test network")

n1 = Node(Name = "node1", Geometry = Point(10, 10))
n2 = Node(Name = "node2", Geometry = Point(20, 20))
n3 = Node(Name = "node3", Geometry = Point(30, 30))
n4 = Node(Name = "node4", Geometry = Point(40, 40))

c1 = Branch(Name = "branch1", Source = n1, Target = n2)
c2 = Branch(Name = "branch2", Source = n2, Target = n3)
c3 = Branch(Name = "branch2", Source = n4, Target = n3)

# add nodes and branches to the network
network.Nodes.AddRange((n1, n2, n3, n4))
network.Branches.AddRange((c1, c2, c3))

# create and fill values of the network coverage
networkCoverage = NetworkCoverage()
networkCoverage.Network = network

networkCoverage[NetworkLocation(c1, 0.0)] = 100.0
networkCoverage[NetworkLocation(c1, 10.0)] = 200.0
networkCoverage[NetworkLocation(c1, 20.0)] = 300.0
networkCoverage[NetworkLocation(c2, 8.0)] = 400.0
networkCoverage[NetworkLocation(c2, 15.0)] = 500.0
networkCoverage[NetworkLocation(c2, 20.0)] = 600.0
networkCoverage[NetworkLocation(c2, 25.0)] = 700.0
networkCoverage[NetworkLocation(c3, 15.0)] = 800.0
  
```

Figure 14: Create network and network coverage (Python)

the gap of a .NET based [NetCDF API](#) that makes use of the features of the .NET platform.

The design of an [API](#) often feels a bit like tightrope walking. There need to be a balance between high level of usability and performance on one side, as well as a balance between completeness versus simplicity.

By following the general guidelines of a domain driven design the [API](#) does adhere to good practices. Whether it is actually a usable one depends on the experience of users. Therefore we invite readers to try out the [API](#) and provide us with feedback and critical comments.

The [API](#) as well as its implementation will be released as an open-source project. Currently a draft version is already available as a branch of a [SharpMap project](#).³⁷

Bibliography

V. Balaji and Z. Liang. Gridspec: A standard for the description of grids used in earth system models. In *Workshop on Community*

Standards for Unstructured Grids, October 2006.

G. Donchyts, S. Hummel, S. Vaneček, J. Groos, A. Harper, R. Knapen, J. Gregersen, P. Schade, A. Antonello, and P. Gijssbers. Openmi 2.0 what's new? In *International Congress on Environmental Modelling and Software*, July, pages 5–8, 2010.

JL Goodall, DR Maidment, and J. Sorenson. Representation of spatial and temporal data in arcgis. *GIS and Water Resources III*. AWRA, Nashville, TN, 2004.

B. Howe and D. Maier. Algebraic manipulation of scientific datasets. *The VLDB journal*, 14(4):397–416, 2005.

H.W.J. Kernkamp, A. van Dam, G.S. Stelling, and E.D. de Goede. Efficient scheme for the shallow water equations on unstructured grids with application to the continental shelf. *Ocean Dynamics*, 61(8):1175–1188, 2011. doi: doi:10.1007/s10236-011-0423-6.

S. Nativi, J. Caron, B. Domenico, and L. Bigagli. Unidata's common data model mapping to the iso 19123 data model. *Earth Science Informatics*, 1(2):59–78, September 2008. doi: 10.1007/s12145-008-0011-6.

B. Nordgren. An iso19123 coverage primer. Integration guide, USDA Forest Service, 2006.

D.J. Peuquet. Time in gis and geographical databases. *Geographical information systems*, 1:91–103, 1999.

³⁷<http://bit.ly/functional-coverages>

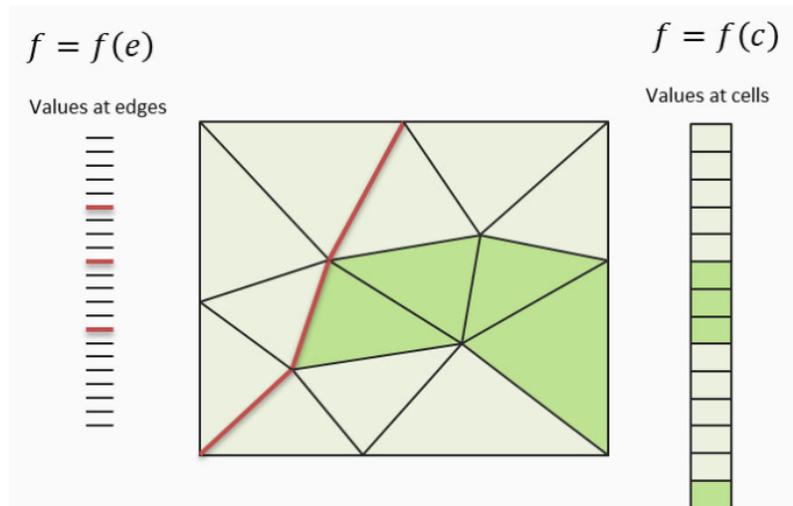


Figure 15: Values defined on unstructured grid cells (faces) or interfaces between cells (edges)

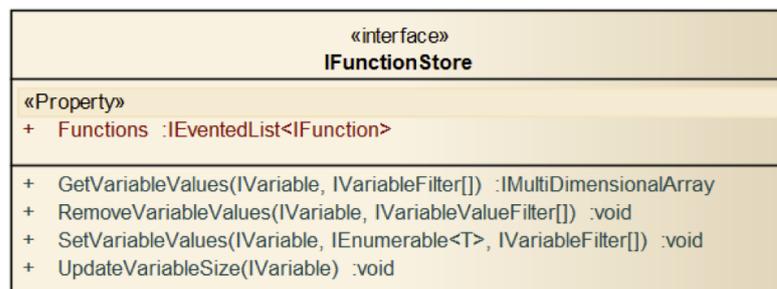


Figure 16: IFunctionStore - repository of functions and their values

R. Rew and G. Davis. Netcdf: an interface for scientific data access. *Computer Graphics and Applications, IEEE*, 10(4):76–82, 1990.

G.K. Rutledge, J. Alpert, and W. Ebisuzaki. A climate and weather model archive at the national oceanic and atmospheric administration. *Bulletin of the American Meteorological Society*, 87:327–341, 2006. doi: 10.1175/BAMS-87-3-327.

W.J. Schroeder, L.S. Avila, and W. Hoffman. Visualizing with vtk: a tutorial. *Computer Graphics and Applications, IEEE*, 20(5):20–27, 2000.

R.P. Signell, S. Carniel, J. Chiggiato, I. Janekovic, J. Pullen, and C.R. Sherwood. Collaboration tools and techniques for large model datasets. *Journal of Marine Systems*, 69(1-2):154–161, 2008.

L. Treinish. A function-based data model for visualization. In *Proceedings of the IEEE Visualization 1999 Conference Late Breaking Hot Topics*, pages 73–76, 1999.

M. Wachowicz. *Object-oriented design for temporal GIS*. CRC, 1999.

GIS Geographic Information System Geographic information – Schema for coverage geometry and functions

Appendix: Definition of Terms

- Feature** Geospatial Feature as defined by OGC. We will call Feature any type that implements at least the Geometry property and has a set of other attributes.
- Geometry** Feature Geometry as defined by OGC. An attribute of the Feature.
- Coverage** Specific type of Feature that can generate a value for any point within its domain. Examples include raster images, a polygon overlay or a digital elevation matrix. In other words, it is a geospatial feature and a function at the same time.
- Function** A vector(-valued) function. A function of one or more variables whose range is n -dimensional. It associates dependent variable (component) values with independent variable (argument) values. As an example, a scalar function has $n = 1$ (a one-dimensional *range*-space), but it may still ‘live’ on a multi-dimensional *domain*-space.
- Variable** Defines a value that may change within the scope of a given problem. The mathematical definition is used, not to be confused with a computer science variable. Variables can be independent or dependent on other variables.
- Unit** Defines unit of measure for variable values. For example [m], [m/s], etc.

Appendix: Acronyms

- OGC** Open Geospatial Consortium
- NetCDF** Network Common Data Format
- UCAR** University Corporation for Atmospheric Research
- API** Application Programming Interface
- GDAL** Geospatial Data Abstraction Library
- CDM** Common Data Model
- OpenMI** Open Modeling Interface
- VTK** Visualization Toolkit

Array In general a multi-dimensional array of some value type.

Gennadii Donchyts, Fedor Bart, H.R.A (Bert) Jagers, Arthur van Dam:

Deltares,

Rotterdamseweg 185, Delft, The Netherlands

P.O. Box 177, 2600 MH Delft, The Netherlands

gennadii.donchyts@deltares.nl

fedor.baart@deltares.nl

bert.jagers@deltares.nl

arthur.vandam@deltares.nl

Opticks Open Source Remote Sensing and Image Processing Software

A Community College GIS Program, and Collaboration

Nathan Jennings

*American River College, Sacramento, California, USA,
nate@jenningsplanet.com*

Abstract

The author has created and taught a remote sensing and digital image processing course as part of the core curriculum in the Geographic Information Systems (GIS) Program at American River College since 2005. Until recently, the course was taught using proprietary digital image processing software suitable for processing remotely sensed imagery. As a result of tough economic times for the college and the Sacramento region, alternative “software tools” were investigated for viable alternatives to use within the remote sensing course. The Open Source Software community was researched and Opticks, an open source digital image processing software for remotely sensed imagery, was chosen because of its useful user graphical user interface, breadth of common image processing functions similar to those found in proprietary image processing software packages, ability to use a variety of remotely sensed image sensor formats, ability to integrate other geospatial information (such as vector data), the ability to continue to customize and enhance the functionality of the software, and strong user and development support.

The author reviewed a number of open source software packages that provide similar functionality to proprietary image processing software packages as well as its relative ease of use and compliments the current material taught in the remote sensing course at American River College. Opticks was found to have the best overall capability for a “free” open source software package that is suitable for a community college introductory remote sensing course.

As Opticks is continuing to be used within the remote sensing course at American River College, the author has formed an informal relationship with Ball Aerospace staff (the developers of Opticks) to assist with creating additional functionality with RADAR data, serve as a Google Summer of Code mentor since 2010, write a functional user guide, and provide feedback to the developers to enhance the software. As a result, the relationship has fostered a mutual benefit to Ball Aerospace, the American River College GIS Program, and for students seeking knowledge and experience in the remote sensing field.

Introduction

American River College is one of four campuses in the Los Rios Community College District in Sacramento, CA and has offered an Associate of Science degree and certificate in GIS for a number of years. The GIS Program has gained worldwide recognition as a high quality higher education program to gain “real world” and hands on knowledge and experience in the GIS and remote sensing field and its application to

many sectors of the local, regional, national, and international business communities. The American River College campus serves as the premiere campus where students come to learn and obtain useful GIS and remote sensing skills as well as critical thinking, problem solving, and collaboration that the job market demands.

In 2005, the author designed and created the Introduction to Remote Sensing and Digital Image Processing course that served to fill a long-time gap in the GIS Associate of Science and GIS Certificate programs. In the initial years of the remote sensing course a proprietary digital image processing software package was used as the primary software tool. In more recent years with tough economic conditions for both the college and the region, alternative software solutions were required to ensure that the remote sensing course remained a challenging and useful course in the GIS Program.

The remote sensing course does not have any prerequisite courses. As a result, students who enroll in the class are often not accustomed to remotely sensed imagery (satellite and aerial imagery), digital image processing, GIS, or any related software tools. Although a number of open source software solutions with long user and development histories are available, many are not tuned to the “novice” software user. When deciding on a high quality open source software solution for this course, Opticks, developed by Ball Aerospace, was chosen.

Remote Sensing and Digital Image Processing Software in the Community College

For many years proprietary digital image processing software (such as ERDAS, ENVI, PCI, and TNT MiPS) has been the “typical” software found in remote sensing courses at higher educational institutions. Remote sensing and image processing principles, concepts, and methods have typically been taught at four year and/or graduate institutions and are often not found in a community college curriculum since these courses are often specialized and are often taken by students in upper division courses. In addition, four year colleges and universities typically have larger budgets to provide appropriate software for teaching these courses over community college computer labs.

American River College began by providing labs with proprietary software and has since had to evaluate other alternatives. Some of the criteria used to choose the Opticks software is based on the following criteria:

- Ease of use
- Functionality similar to professional grade remote sensing software
- Comparable interface to other GIS and/or remote sensing software
- Available documentation to use the software
- Ongoing development support from the parent organization

A number of open source software packages exist that include remote sensing and digital image processing capability (such as GRASS, OpenEV, and OSSIM) have either a high learning curve or relatively limited functionality without additional compiling, installation of libraries, or add-on modules. For an inexperienced student in remote sensing, GIS, and image processing software and techniques, using some of the open source software can be overwhelming. As described below, Opticks provides a full range of common image processing functions as well as more advanced functions such as those related to feature identification and hyperspectral analysis while being easy to use and Opticks continues to add and improve the functionality in their software.

Remote Sensing at American River College

The remote sensing course in the GIS program at American River College includes topics related to remote sensing principles and digital image processing techniques. Many students are looking for practical experience in remote sensing and image processing that can be used in a variety of fields. Two four year universities in the Sacramento region, UC Davis and Sacramento State, offer a number of classes in remote sensing and image processing and focus more on principles and theory versus practical use of the software. The remote sensing course at American River College was created as a survey course so that the student can obtain the fundamentals in remote sensing theory as well as obtain actual “hands on” experience with the tools (i.e. software) to analyze remotely sensed imagery. Knowledge areas covered in the remote sensing course are:

- Image fundamentals
- Electromagnetic spectrum and remote sensor sensitivity
- Remote sensors (airborne, spaceborne, RADAR, LiDAR, hyperspectral)
- Image enhancement
- Image classification (traditional unsupervised/supervised/hybrid)
- Specialized image processing techniques such as texture analysis, image rectification, topographic surface creation, target detection, and feature based image analysis

Opticks

The Opticks software was developed by Ball Aerospace in conjunction with the US Air Force in 2000 primarily to conduct hyperspectral image analysis (Streithorst and Considine, 2011). Over the next several years Ball Aerospace and the US Air Force provided the core Opticks code as open source software in 2007 (Streithorst and Considine, 2011). Since then, a number of fundamental software improvements and enhancements have been made.

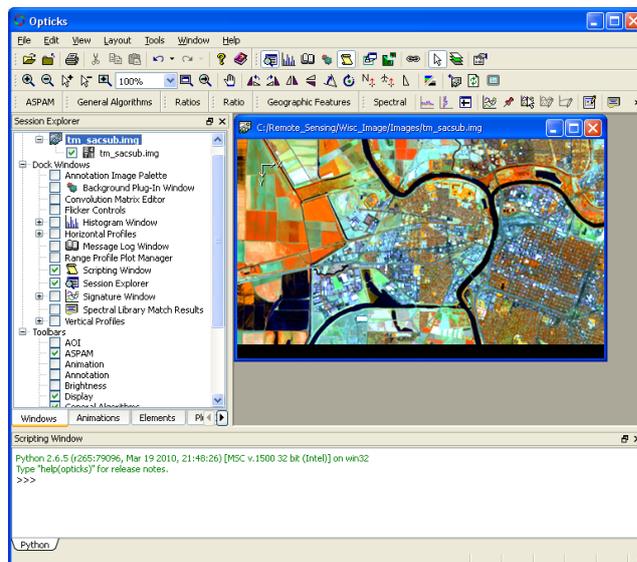


Figure 1: Opticks software interface showing an example Landsat TM (Thematic Mapper) image and Python scripting window

Figure 1 shows a typical interface that the image analyst can use to process imagery. A portion of a Landsat TM (Thematic Mapper) image is displayed. Landsat is one of many satellite based remote sensors available for image processing. The right side of the Opticks software provides the ability for the user to add additional toolbars to the interface as well as additional tools such as histograms and a spectral signature window to explore the details of the image (e.g. bands, rows, columns, wavelengths, changing the display bands, data type, etc). Figure 1 also shows the Python scripting window that can be turned on where users can directly add Python code to process images using custom algorithms and the Opticks Python module. In addition, users can run completed Python scripts from this window or through the Wizard Builder to create a button to run a script.

Opticks core functionality include many common image processing tasks including, image subset, image filters, band ratios, and band math (e.g. arithmetic computations between bands). Opticks also includes the ability to create image processing workflows using the Wizard Builder. Some of the Wizard Builder Processes can easily become part of custom built toolbars and menu options. Advanced options include unsupervised and hybrid image classification and a series of tools to process hyperspectral imagery. Additional functionality has been provided through students participating in the Google Summer of Code Program and include feature extraction, RADAR processing, and astronomical image processing.

The Use of Opticks in a Community College GIS Program

The Opticks software is a primary digital image processing software for the remote sensing course that is part of the GIS Associate of Science degree and GIS Certificate Program at American River College. As a result of smaller college budgets and a long-term down economy in Sacramento, funding

for commercial software is more difficult to fund. In addition, proprietary software companies are reluctant to provide “free” fully functional student versions of their software to students who are pursuing degrees and skills that use their software. Since the remote sensing course does not require any pre-requisite courses and students are often unfamiliar to image processing and GIS software tools, a “free” and open source and intuitive software, such as Opticks, is an appropriate choice for students who are introduced to both challenging content and digital image processing protocols. The financial burden to students and to the school is reduced and students can install the software on personal computers so they have ample opportunities outside of the classroom to learn and experience the methods and analytical tools to process remotely sensed imagery.

Opticks has served as an integral part of the remote sensing course offered at American River College. The software contains many fundamental image property and processing tools for a variety of remotely sensed image formats which are taught as part of the course. Fundamental image property tools include image format, number of image bands, rows, columns, pixel type, data type, wavelength characteristics, band histograms, changing the band combination display, etc. Fundamental image processing tools include image contrast, filters, resampling, mathematical expression development between bands (such as arithmetic, square root, powers, and a variety of mathematical functions). The Wizard Builder allow students to build custom multi-step processes, such as unique band ratios to analyze biophysical characteristics. Normalized Vegetation Difference Index (NDVI), Tasseled Cap, water stress, vegetation stress, biomass quantity, etc are a few examples of common algorithms that can be taught and built using the graphic process wizard without using any special programming methods (Figure 2).

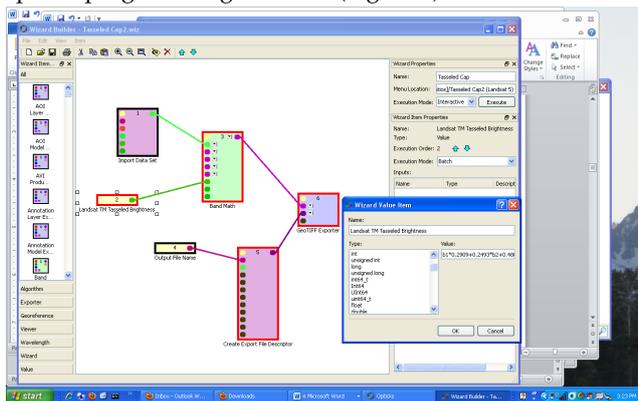


Figure 2: Opticks Wizard Builders showing the Tasseled Cap algorithm for Landsat 5 imagery

Opticks can also be used for more advanced image analysis such as spectral signature development and evaluation and image classification (supervised and unsupervised methods). Students learn about creating, evaluating, and using spectral signatures as part of the image classification process. The remote sensing course also includes a segment focusing on hyperspectral imagery (images that often contain more than 200 specific wavelengths). Opticks includes a series of specialized tool for processing hyperspectral imagery. These include removal of image anomalies, material identification, spectral library comparisons, atmospheric correction, spec-

tral signature creation, and materials identification. A recent enhancement to Opticks has been a feature identification method that was created by a Google Summer of Code participant in 2010. Many of these functions are provided through the Spectral toolbar and extension (Figure 3).

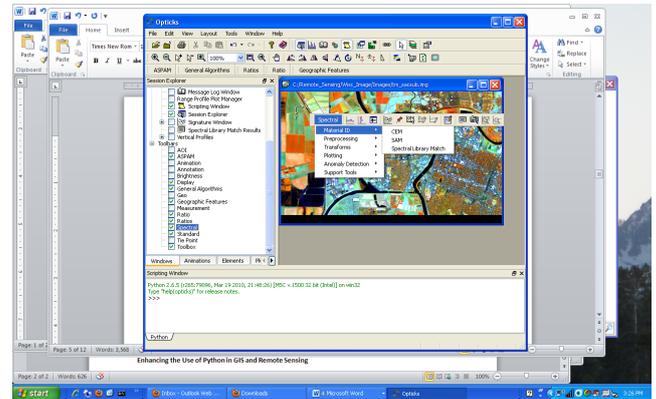


Figure 3: The Spectral toolbar for Opticks showing a variety of spectral analysis capabilities

The author has also produced a functional user guide for Opticks that students can use to begin learning the Opticks software and that can augment the Opticks Help (Figure 4). The user guide is one of several documents the author has written and provided to students in the remote sensing course.



Created by: Nathan Jennings
www.jenningsplanet.com
 Created on: 08.21.2010
 Updated on: 11.14.2010

Figure 4: Opticks User’s Guide created by the author

RADAR Algorithm Development Using Opticks

The author recently created a series of speckle reduction (image smoothing) and texture analysis (algorithms that measure how similar or different groups of pixels are from one another in the image) algorithms that are tailored to RADAR (Radio Detection And Ranging) remotely sensed imagery. In a separate Google Summer of Code 2010 project, a student also created some additional RADAR processing algorithms. Both sets of algorithms were developed using the Opticks C++ libraries and the C++ programming language. Once a code developer learns some of the basic constructs provided by the Opticks development team, developing specific image processing algorithms is relatively straight forward, provided the programmer has a background in image or signal processing, and a strong background in C++. Many algorithms tailored for remotely sensed imagery can be found throughout image processing textbooks and peer reviewed articles in professional journals.

The author teaches a segment on RADAR image processing in the remote sensing course. Since the author has a background in RADAR image processing and Opticks did not have this functionality built into the core software, the author was able to develop several fundamental RADAR processing routines that were suitable for an introductory remote sensing course. Figure 5 shows a screen shot of the Opticks interface with the custom RADAR Processing module. The image in the viewer is from a Canadian RADAR satellite, RADARSAT-1.

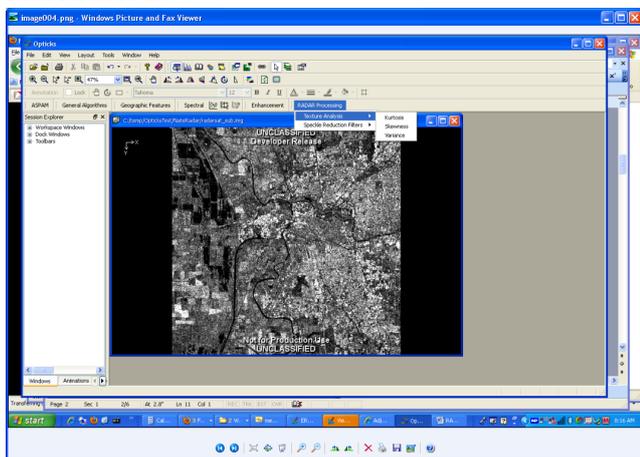


Figure 5: Opticks showing the custom RADAR Processing module showing texture analysis

Anyone can develop or expand the functionality of Opticks using the Opticks libraries and the C++ programming language. Currently, professional grade compilers (such as Visual Studio) and the free version (Visual Studio Express) are supported for Windows development. Sun Solaris (Sun Studio) compilers are supported for Sun Solaris operating systems. Code developers should have a strong command of C++, object oriented programming, and digital image processing experience. Ball Aerospace staff has a good reputation of providing constructive feedback and examples for those beginning to learn the Opticks objects and include both a developer and user forum.

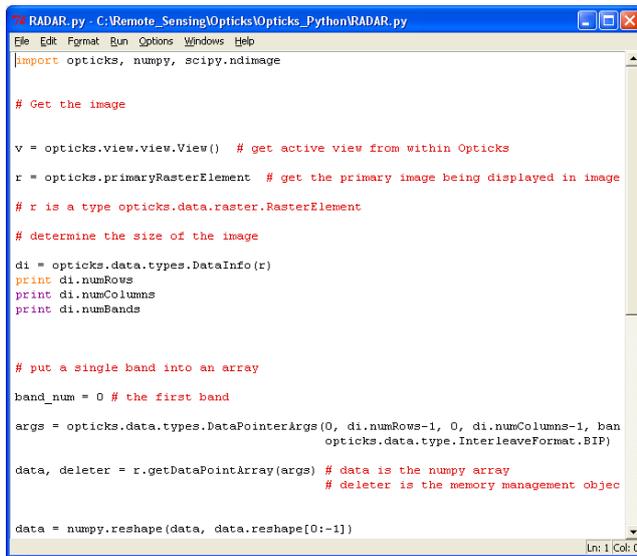
Collaboration with Opticks and the Google Summer of Code Program

The Google Summer of Code Program is an open source coding program sponsored by Google. Open source coding project organizations (such as Ball Aerospace's Opticks project) can apply to sponsor one or more "students" to assist with expanding the capability of the organization's open source project. To support the Opticks Open Source effort, the author has participated as a mentor since 2010 in the Google Summer of Code Program. In the past two years several students have participated in the Google Summer of Code for the Opticks open source project to make contributions to RADAR image processing, feature detection, and astronomical image processing (extra-terrestrial imagery focusing on planets, stars, and other space phenomena). The author has served as a mentor to the efforts with respect to RADAR imagery since he has requisite knowledge and skill to evaluate RADAR algorithms and image processing results. Ball Aerospace staff serves as the primary leads and provide technical experience with Opticks fundamentals and coding strategies.

Opticks Python Extension

Recently, the Opticks development team has provided an add-on extension to use Python to develop image processing algorithms. If a code developer does not need to develop a custom interface or modify the Opticks core functionality, the Python extension is likely an easier method to develop custom image processing algorithms than developing C++ code. Code developers do not have to "recompile" their code for each new major release of the Opticks software and the deployment of the algorithms can be easier to install for novice Opticks users. Being both a full time GIS professional for the City of Sacramento and an Adjunct Professor at American River College, the author does not have significant time to develop and redevelop custom algorithms for each release of Opticks. Python is a more suitable development environment for the author. The author is currently working on writing the RADAR algorithms Python and the Opticks Python module that were originally developed in C++ as well as some additional algorithms suitable for the remote sensing course. Figure 6 shows a portion of the script being developed by the author for the RADAR module.

A set of Opticks Python modules are provided by the Opticks team for Python developers to read and write image formats as well as gain access to specific characteristics of the image (bands, rows, columns, data type, pixel type, wavelengths, etc). Being able to access these characteristics allows code developers to create special image processing routines that can process imagery on a pixel by pixel basis or performing mathematical expressions between the image bands. In addition, with the use of additional Python extensions (such as NumPy and SciPy) some statistical expressions are already available that may not require special manipulation of the Python code for some algorithms. As a result, the Opticks Python extension allows for a full range of additional image processing capabilities without the significant programming overhead and knowledge C++ often requires.



```

RADAR.py - C:\Remote_Sensing\Opticks\Opticks_Python\RADAR.py
File Edit Format Run Options Windows Help
import opticks, numpy, scipy.ndimage

# Get the image

v = opticks.view.view.View() # get active view from within Opticks
r = opticks.primaryRasterElement # get the primary image being displayed in image
# r is a type opticks.data.raster.RasterElement

# determine the size of the image
di = opticks.data.types.DataInfo(r)
print di.numRows
print di.numColumns
print di.numBands

# put a single band into an array
band_num = 0 # the first band
args = opticks.data.types.DataPointerArgs(0, di.numRows-1, 0, di.numColumns-1, band_num,
opticks.data.type.InterleaveFormat.BIP)
data, deleter = r.getDataPointArray(args) # data is the numpy array
# deleter is the memory management object

data = numpy.reshape(data, data.reshape[0:-1])
Ln: 1 Col: 0

```

Figure 6: Sample Python script using the Opticks libraries

Summary

In the past the author has used ERDAS Imagine as the primary software in the remote sensing course. ERDAS imagine is a fully functional professional image processing software package. Much of the functionality found in ERDAS is also found in Opticks. Both provide basic image processing functionality such as image enhancement, filters, ratios, image subset, and image construction. ERDAS and Opticks support a multitude of image and sensor formats and data types. Both provide more complex image processing routines such as hyperspectral analysis and feature based image processing. Both packages do have a “modeling” capability, although the functionality in Opticks is not as straight forward as that found in ERDAS. The Opticks software user must become more familiar with image elements (bands, rows, columns, header files, data types) than required with ERDAS. The current version of Opticks does not support the use of LiDAR data or the construction of topographic surfaces from raw LiDAR data. Also, Opticks only supports a single unsupervised method (K-Means) and does not have a fully functional “supervised” classification workflow. Opticks does have a more refined set of methods for hyperspectral and target identification than that of ERDAS. Despite, some of the lack of functionality of Opticks compared to a proprietary software such as ERDAS, for being a free, open source software, these short comings can be accommodated with the use of ESRI’s ArcGIS software, which American River College has a site license. Opticks can nearly support the entire knowledge base provided to students in the American River College remote sensing course.

Opticks open source digital image processing has been an

integral part of the remote sensing class in the American River College GIS degree and certificate programs. Using Opticks has helped save funding for the school and useful and “free” software for students while being able to maintain a high quality educational experience to students who want to gain knowledge in the field of remote sensing and digital image processing. Opticks has been a good choice for an educational program, especially for those students who often enroll in the remote sensing course with little or no background in digital image processing, remote sensing principles, or GIS. The development community at Ball Aerospace as well as external contributors is strong and provide routine updates and improvements that are typically released a couple of times a year. In addition, with the Opticks open source project in the Google Summer of Code Program additional enhancements have been able to be achieved in a timelier manner than developing them completely in-house by Ball Aerospace. The author is appreciative to be able to contribute and use to a strong collaborative project like Opticks and that a free and open source solution exists that can be used in a community college GIS program.

The Opticks software and project can be found at <http://opticks.org>. The American River College GIS program can be found at <http://wserver.arc.losrios.edu/~earthscience/gis.htm> and specific topics in remote sensing and the course taught at American River College as well as a copy of the Opticks User’s Guide can be found at the author’s website³⁸ or can be contacted via email at the above address.

Acknowledgements

The author would like to thank Kip Striethorst, Trevor Clarke, and Michael Considine, and all of the Opticks team for providing Opticks to the open source community and allowing him to participate in the development of the Opticks software.

References

- ERDAS, Inc. Corporate website, viewed December 12, 2011, www.erdas.com.
- Exelis Visual Information Systems, Coporate website, viewed December 12, 2011, www.exelisvis.com.
- Open Source Software Image Map, website, viewed December 12, 2011, www.ossim.org.
- OpenEV, website, viewed December 12, 2011, www.ossim.org.
- PCI Geomatics, viewed December 12, 2011, www.pcigeomatics.com.
- Striethorst, K. and Considine, M., Ball Aerospace 2011, History of Opticks, viewed July, 2011, <http://opticks.org/confluence/display/opticks/History+of+Opticks>.

³⁸<http://www.jenningsplanet.com>

Implementation, challenges and future directions of integrating services from the GIS and decision science domains

A case of Distributed Spatial Multi-Criteria Evaluation

Luc Boerboom

Faculty of Geo-Information Science and Earth Observation (ITC)
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands, boerboom@itc.nl

Özgün OskayAlan

Faculty of Geo-Information Science and Earth Observation (ITC)
University of Twente, alan@itc.nl

Abstract

We are implementing an open source project for spatial decision making called Distributed Spatial Multi-Criteria Evaluation (DSMCE) under an EU project for inter-regional development on forestry and climate change adaptation (ForeStClim).

In this paper, we first describe what DSMCE is and what it does. We have designed an extensible architecture for integrating services of two domains, respectively the GIS and Decision Sciences domain. Thereby we delegate domain expertise to available implementations. We use the Service Oriented Architecture (SOA) paradigm to build our DSMCE service and application. Integration is implemented by the use of open specifications and protocols coming from these domains. DSMCE is not only extensible in terms of the external services it uses, it also is extensible as an application because it is developed with OSGi technology which that brings advanced modularity.

Second we share observations about implementation challenges we have addressed. These challenges are related to the design of integration of the two domains, the ability of specifications to address real implementation problems, and the reliability and quality of available open source tools. These lead us to conclusions about the solutions we had to implement.

Third and finally we give an overview of future directions. Some of these topics relate to the spatial domain, e.g. the use of Web Processing Services (WPS) for pre and post processing around decision analysis, others to the decision sciences domain, e.g. the integration of other non-spatial data sources and services, or collaborative decision making.

Introduction

If spatial multi-criteria evaluation (SMCE) (Herwijnen, 1999, Sharifi and Retsios, 2004) methodologies, implementations, and expertise could evolve on the web, both decision making and the decision aiding methodologies could develop in new directions. SMCE in desktop applications has been used in analytical academic or consulting studies such as transport (Sharifi and Boerboom, 2006, Keshkamat et al., 2009),

environmental management (Zucca et al., 2008) in poverty assessment (ODPM, 2004, Baud et al., 2009). It is also used in participatory decision processes either normatively motivated because it should make decisions more democratic, or rationally motivated because it should make decisions more informed, or instrumentally because it should make decision responsibility for possible failures shared (Stirling, 2006). The number of publications has experienced a strong increase (Malczewski, 2006).

But on the web it could aid and change individual and collaborative decision making. It could create need and opportunity for expanding the range of methods for the analysis of spatial preference of groups of people, for collaborative analysis of conflict and consensus, and for learning about decision making and decision making processes. It could add value to and between spatial data infrastructures, and perform integrated assessment between organizational mandates. And it could get infrastructural properties (Boerboom, 2010).

So far there has only been one implementation of SMCE that is server-based with the lightness of a browser client, which is ParticipatoryGIS.com (Borouhaki and Malcewski, 2010), but unlike our implementation it has not been implemented as a generic tool but for a specific project nor does make use of OGC web services standards and implementations.

We present the first prototype implementation of the open source distributed spatial multi-criteria evaluation (DSMCE) web application. It is distributed, not only in the concept of distributed computing, because it can collect data from distributed data sources, i.e. Web Feature Services (WFS), or, if data cannot be exchanged because of data policies, data value, bandwidth, and other reasons, it can be distributed to these data sources and collect only the intermediary outputs. Also, decision makers are geographically distributed or in time. And development of it can be distributed, given the open source nature, provided good programming practices and systems are maintained. Distributed SMCE is an open source web application development project hosted on the source forge Kenai.³⁹

Implementing DSMCE with spatial OGC standards was not as straight forward as the intended by OGC (Percivall, 2010). We did not find a systematic review of issues around implementations and use of OGC standards in literature, although (He et al., 2009) address one of the issues using multi-version WFS'. It is beyond the scope of this paper to do so. But we consider it useful for future development of standards and implementations to explore the challenges we have faced and the solutions we have developed, which could become part of future more systematic studies.

So the outline of this paper is as follows. We first briefly describe the particular use case for which this web application

³⁹<http://kenai.com/projects/distributed-DSMCE>

is being developed and its generalization. Second we describe the web application. Then we discuss implementation challenges with open source modules. Finally, we describe future directions.

A specific use case and its generalization

The specific use case is the following. Most European forest organizations have included climate change in their strategy documents. Now these strategic intentions need to be translated to adaptation plans in the different regions. This occurs as part of the regular forest management adaptation planning processes. Therefore spatial evaluation of vulnerability and adaptive options needs to be considered in these processes. In the ForeStClim project on “Transnational Forestry Management Strategies in Response to Regional Climate Change Impacts”, within which the distributed spatial multi-criteria evaluation web application is developed, we intend to compare several regions. But rather than applying a uniform evaluation approach we recognize the regional variability.

So the problem is that evaluation of climate change vulnerabilities and suitability of adaptive option is distributed. There are different forest management organizations in the different regions in North-western Europe. Each organization works in its specific forest policy environment and has specific policy objectives. And they all have different data, technical data environments, and data policies. To summarize, decisions of these foresee management organizations are idiosyncratic in only partly shared policy and market environments, and there is neither value in making databases and datasets interoperable nor in semantically harmonizing data.

Finally, the project hopes that in the exchange of different evaluation approaches the different regions will gain ideas to improve their own understanding of vulnerability and adaptive options. And that these can be communicated to European policy bodies as the Ministerial Conference on the Protection of Forests in Europe

Generalizations from this specific use case are the following. As far as decisions are concerned, DSMCE can be used for decisions where data can be the same for different decision makers, but decision makers can partially or fully use different data and data sources on internet or intranet. Also it is meant to communicate preference structure on Internet for decision makers to learn from each others’ approaches to evaluation and integrated assessment. Often data cannot be shared or made interoperable because it is just not worth the effort since use of data is infrequent and/or idiosyncratic, and does not pay off the effort of making databases interoperable. Also data can be too valuable to be shared or outdated data should not be used. DSMCE supports decisions with organizational databases and with data infrastructures (SDIs) but also at the fringes of SDIs.

DSMCE

First we describe what DSMCE does. Then we describe its architecture. Finally we list functional and technical innovations. DSMCE consists of two parts, the web application (frontend service) and the backend service.

The web application (Figure 1) opens in any web browser. It initially is a single screen with four panels. Spatial multi-criteria evaluation takes place in the central panel. Here objectives and criteria can be structured, standardized with maximum standardization to a 0-1 scale, prioritized with the expected value ranking method, and aggregated (Nijkamp, 1990, Sharifi and Retsios, 2004). The Data panel (left) provides access to web feature services. From the service the user receives some technical information and a list of layers offered. After selecting a layer, its attributes will be listed as thumbnail maps and their attribute names. Multiple thumbnails can be popped up (pop up window) to an image and some key statistics of mean, standard deviation, minimum and maximum value in order to obtain a feeling for the data. The thumbnails can be dragged and dropped into the maps field. The spatial view panel (to the right) is a viewer that uses OpenLayers, and shows the transparent output map of well (green) and poorly (red) performing areas on top of base layer. Here it is OpenStreetMap. Clicking on one of the polygons, attribute names and values for that polygon will be displayed in the Spatial Info panel (lower right corner).

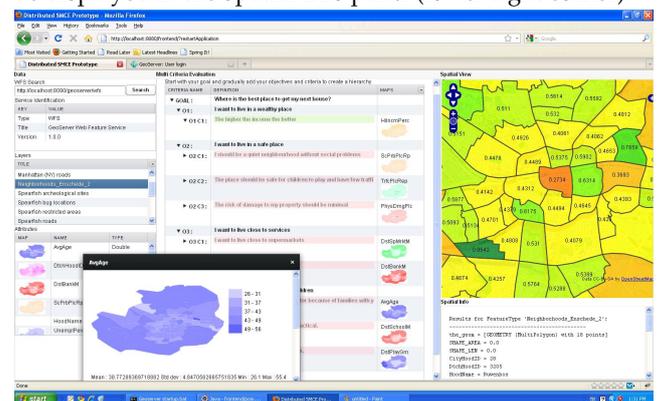


Figure 1: Screenshot of the Distributed Spatial Multi-Criteria Evaluation web application

As depicted in Figure 2 the architecture consists of three main components: Spatial Services, Decision Support Services and finally the D-SMCE service. Spatial Services consist of data services such as WFS and data processing services such as WPS. In the decision support service component we have the Decision Deck as a multi-criteria aiding service. D-SMCE itself is a client of all those external services and plays a role of a mediator service to bridge two domains (Spatial Domain and Decision Support domain) to perform required tasks of a Spatial Multi Criteria Analysis. It is implemented on the Java platform and uses modular design following OSGi specifications.

If we look inside the D-SMCE component there we have two services as well. We have the backend service. In this backend service we implement data access and business layers. Here we have modules of several client implementations to access external services for data retrieval and data processing. Also we have some other modules for required calculations that external services cannot offer. Then there is the frontend service. This service consists of the presentation layer together with some utility modules that are needed by the web application (e.g. user profile management, access control). The presentation layer is implemented with the Vaadin framework.

Our idea is to clearly separate these two services, backend and frontend, by running them as two independent services. If a necessity arises, we may like to add a new component to this architecture such as statistics service by implementing a required client in the backend service, as it is depicted in the bottom of the figure.

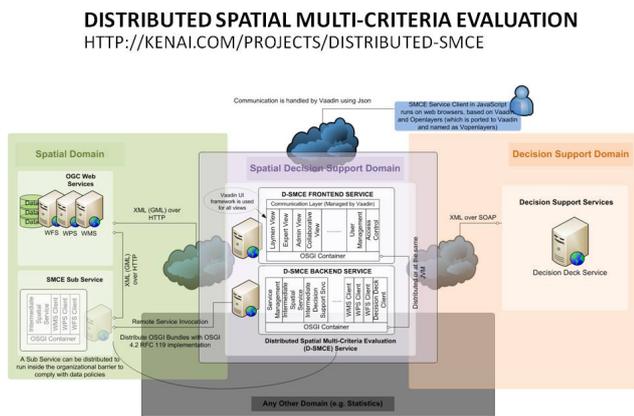


Figure 2: Architecture of Distributed Spatial Multi-Criteria Evaluation

The core of this new web-application is functionally and technologically innovative. Functional innovative aspects are:

- SMCE on the web. Spatial data is currently shown and sometimes downloadable on internet. A map can be looked at one at a time. With DSCME multiple maps can be viewed, interpreted, and aggregated to perform spatial evaluation.
- Integration with MCDA services. (fig 2). We will apply non-spatial multi-criteria decision analysis web services standards (XMCD) (Decision Deck, 2011) in the spatial domain. Weighted summation MCE is the first method but once embedded others can be added.
- Spatial data from different locations on internet can be brought to the web-application
- Distributed calculation. If data providers do not want data to be moved, parts of the backend of the DSCME can be served from different locations where parts of a criteria tree can be analyzed in different web locations and results shared to a "central" location.

Technological innovative aspects are:

- Extensibility of services. DSMCE is a service oriented platform that mediates the interaction between spatial and non-spatial services available on the web. This feature is unique since the majority of decision support software are either desktop or isolated web applications. This feature opens opportunities for extension of the system (such as adding statistical capabilities, etc.)
- Open standards and communities. DSMCE uses open standards developed by communities. It uses OGC Standards (Web Feature Service, Web Processing service, and Web Mapping Service) to process spatial data and Decision Deck standards for the non-spatial multi-criteria decision analysis (MCDA) methods. The underlying philosophy is to delegate domain expertise to other implementations that

are represented on the web (i.e. as web services) and build a reliable, extensible infrastructure for the mediation of all these delegated services. Therefore comparing to other software systems in the field of spatial decision making, Distributed SMCE can be positioned as a framework rather than a tailor-made application.

- Use of OSGi technology. (fig. 2) OSGi technology is a specification to create modular applications in the Java platform. The choice of OSGi technology for DSMCE has been a key decision. It is technically and administratively motivated. Technically, the extensibility that is gained by the use of Web Services cannot be utilized without a modular web application that mediates interaction between different kinds of web services. Possibly more important is the administrative motivation of distributed partial analysis within different organizational boundaries. Such scheme requires both modularity and convenient tools for distribution of logical components. Distributed SMCE can follow this scheme by using OSGi.
- Contribution of own services. Distributed SMCE is not only mediating web services, but also performing intermediate computations which are not available on the web or too specific to be standardized under OGC, Decision Deck or other standards. For instance it extracts relevant information from one service, e.g. meta-data and certain descriptive statistics from maps like the maximum and minimum values, and uses this information for other services such XMCD) services. Although the computation might be specific still it may have demand as a module or as service on the web. Also for these kind of use cases, OSGi technology and the current architecture of DSMCE gives enough room and flexibility.
- Server side rapid development with Ajax/Vaadin. The client, i.e. User Interface (fig. 2), is based on the Vaadin Server Side Ajax UI Framework. Like all other Ajax frameworks Vaadin provides rich user experience. Vaadin has some advantages compared to other frameworks in terms of Rapid Application Development. This is essential for an open source project where 3rd party developers of Distributed SMCE would like to extend Distributed SMCE and so its web client. The choice of Vaadin has some important implications. First, since Vaadin is a server side Ajax framework, it has a fairly 'thin' User Interface layer that runs on web browsers of the end users. Second, browser compatibility issues are handled by Vaadin. So the developer does not have to worry whether the developed code is working with different browsers. Third, the big majority of operations, communication and security is handled in the server on a Java platform. This feature of Vaadin gives us opportunity to develop a good degree of modularity in combination with the modularity enforced by OSGi that applies to the Java platforms. Finally, since most of the current web applications are heavily based on Javascript, it is hard to modularize them by using specifications such as OSGi, or frameworks developed for imperative programming language platforms such as Java. We have chosen Vaadin because it minimizes the use of Javascript.

Discussion

With the prototype we aim to explore possible limitations and future challenges. We for instance need to experiment with performance. Some design and implementation issues we have already encountered and addressed.

Design Issues

Two major design choices were made. The first was to make the web application highly modular and distributable. The second was substituting WPS for Decision Deck web services.

Data exchange restrictions

Two challenges in the design were to handle the size of geographic data layers and the assumption that data policies require data to reside inside organizational boundaries. Both challenges violated our initial design where we wanted to rely fully on external services. But now the application had to become distributable as well in cases data services could not directly be provided. So these challenges required that the application at least partially be local (insider the organizational boundary) and created the need to design a distributed application using distributed services.

As a solution to these challenges we consider the use of meta-data and delegating the analysis (e.g. calculation of descriptive statistics by WFS) to the data services wherever possible. And for further analysis on the dataset within organization boundaries we deploy a utility service. This service can perform analysis and then can transmit partial and intermediary results to the main system.

Using OSGi technology gives this possibility of designing a highly modular application. Another advantage are its "Remote Services" (OSGi Alliance, 2011) for distributing modules over the web. We aim to use "Remote Services" of OSGi to be able to distribute our necessary modules across organizational boundaries to give us opportunity to retrieve only meta-data and partial and intermediary results to finalize the analysis.

Finding proper service for decision aiding

At the start of the project we considered to implement the multi-criteria decision aiding components as Web Processing Services (WPS). However, we encountered several challenge about WPS in relation to our project. One challenge was already formulated by (Friis-Christensen et al., 2007) as the absence of "separation of geometry and attribute data: Geometry information, though not required for a large number of processing operations (like classification and attribute normalisation) is dragged along as information ballast slowing down the performance of applications. Examples for specifications looking into this issue are the related OGC discussion papers on the Geolinking Service (OGC, 2004b) and the Geolinked Data Access Service (OGC, 2004a)." Indeed in our application the majority of use cases required only attribute data of features to be processed and analysed and the geometric information served just the mapping. These attributes are, partially in pre-processed form, input to XMCDAs web services. So we abandoned our initial idea to implement all analysis as WPS.

Another reason to abandon WPS was the lack of concepts to provide semantics of the multi-criteria decision aiding do-

main. This problem of a lack of concepts to provide semantics was already observed by (Foerster and Stoter, 2006). This issue together with the fact that WPS provides very generic interface, would have required us to spend considerable effort to implement decision aiding algorithms for spatial data in WPS.

Therefore we looked at an alternative solution where we could separate analysis of attribute data from analysis of and operation on geometry. And we could also find web services that provide decision semantics. We found a solution in the decision sciences domain where open standards for multi-criteria decision aiding (MCDA) web services have recently been developed in the Decision Deck Project. Now we only consider WPS for truly geometric pre and post processing operations such as overlay analysis or calculation of spatial metrics as criteria.

Finally, we abandoned WPS for decision aiding algorithms because of the possibility to process large volumes of data because many criteria maps may be involved and more advanced decision aiding methods are more complex and resource consuming too. As discussed by (Michaelis and Ames, 2009) in such situations it can be more efficient to perform the processing locally.

Implementation Issues

We have faced three implementation issues. First, we have had to address inconsistencies between schema and schema instances. Second we have had to address lack of information about the data. Here we do not mean so much meta-data, but descriptive data. And third we missed open source GIS toolkits with proper documentation.

Inconsistencies between schema and schema instances

We noticed that retrieval of data (or meta-data) becomes fragile because of difference between WFS schema and WFS schema instances or because of missing schemas while retrieving complex types. Since the data retrieval in DSMCE is made from remote WFS servers, which we do not have control of, human errors or bugs in OGC Service implementations can cause bad user experience.

As a solution, we implemented parsers that use a domain model which is a collection of Java objects based on OGC Web Service Common and Web Feature Service specifications. We implemented with Apache Commons Digester (Apache Commons Digester, 2011) library. Although this approach requires implementation of java objects following the domain model, it gives nice flexibility and more tolerance for errors. For retrieval of the data we preferred to use GeoJSON format since it is a lighter format comparing to GML and accessing attribute data is easier.

Data ambiguity

Another main problem is the lack of support for units of attributes and descriptions of attributes. In GML3, schemas for units are defined (Cox et al., 2002), however this information is not being used by the available online Web Feature Services yet. But decision makers will need to know such attribute information and the currently supports name and type elements are not sufficient for a decision maker. Moreover, most

of the time, the name field is cryptic and not explanatory either.

We have not addressed this issue of data ambiguity in the current prototype. For those cases where WFS are used that the user has no control over, we will be providing annotation tools to the decision maker (end user) units and data descriptions assuming the user has other means to obtain units and data descriptions. For those cases where users of WFS are in the same organization as the WFS supplier, custom solutions could be made to create unambiguous data interpretation.

Lack of WFS support for descriptive statistics about the data

The general user of WFS, but certainly the user of WFS via DSCME, needs descriptive statistics such as the maximum, minimum, mean, and standard deviation. For instance of the standard deviation of a certain attribute is small, it has little discriminatory value between the alternatives. Particularly in spatial data sets where the number of alternatives (points, lines, polygons, cells) can be very large, descriptive statistics are very important. So although a decision maker may find the criterion that uses the attribute important, if it is not discriminating, it might even be discarded altogether. But also for properly styled WMS visualization, not only in DSMCE, maximum and minimum values are important.

We expected more support from OGC data services (WFS, OGC). Descriptive statistics are optional services in section 13.3.2 of WFS 1.1.0 specification (Vretanos, 2005): "The schema of the Filter Capabilities Section is defined in the Filter Encoding Implementation Specification. This is an optional section. If it exists, then the WFS should support the operations advertised therein. If the Filter Capabilities Section is not defined, then the client should assume that the server only supports the minimum default set of filter operators as defined in the Filter Encoding Implementation Specification."

Since these descriptive statistics are crucial for decision making, we implemented a simple statistics facility by fetching the whole feature set and using Apache Commons Math library to compute descriptive statistics. We considered checking if a filter is available from a WFS in the capabilities response, but this burdens our system with complexity of checking and error handling. Although we have a working solution, it breaks with our initial idea of using the meta-data and capabilities of external services prior to the core analysis.

Lack of lightweight open source GIS toolkits with proper documentation

As described under section design issues, the majority of operations in DSMCE only use attribute information, not geometric information. However in the available open source GIS toolkits, data structure designs are naturally affected by the traditional structure of GIS data where features are a composition of geometry and attributes. We also noticed many interdependencies between libraries and lack of documentation about dependencies. So it becomes really hard to use toolkits for our lighter needs and we did not find a lightweight GIS toolkit which is efficient for attribute data and helpful for simple mapping. These difficulties and poor documentation motivated us to implement lightweight OGC service clients for WFS and WMS. For that purpose we used Apache Commons HTTP (Apache Commons HttpClient, 2011) library and

to be able to create POST requests with XML encoding we used WAX library for JAVA (Volkman, 2011).

Future directions

Since we have finished only a prototype so far, a lot can still be done:

- First, we want to modularize our systems with OSGi and run it in an OSGi container.
- Second we need to create state persistency, user profiles and workspace.
- Third, we want to proceed with the integration of Decision Deck to provide a good amount of multi-criteria decision aiding algorithms.
- Fourth, we want to add preprocessing WPS so that users can create a suitable criterion map from a geometry of another map or of geometries of different maps.
- Fifth, we have not addressed the issue of discovering data but evaluation and use of OGC Catalogue Service is in our agenda. If DSCME becomes an application that runs within organizations it will need to be customized to use the organizational catalogue.
- Sixth, because of the challenges in data formats and available tools that support Web Coverage Service (WCS) we started our project with vector support. However we know of very good experience of usefulness of raster-based SMCE with the SMCE module we developed earlier in the desktop ILWIS GIS (52North, 2011) and would like to include a raster version in the agenda.
- Seventh, we want to develop the potential for collaborative decision making and explore new decision aiding algorithms.
- And finally, we know chaining services and managing it by the use of workflow managers is very interesting. To be able to satisfy different and complex scenarios in decision making we would like to develop a workflow mechanism for our application. We believe prior to that we need a good degree of modularity in our application.

Conclusions

We have presented a prototype of distributed spatial multi-criteria evaluation web application, which integrates OGC and Decision Deck web services, thereby delegating functionality to the respective expertise domains. It is distributed, not only in the concept of distributed computing, because it can collect data from distributed data sources, i.e. Web Feature Services (WFS), or, if data cannot be exchanged because of data policies, data value, bandwidth, and other reasons, it can be distributed to these data sources and collect only the intermediary outputs. Also, decision makers are geographically distributed or in time. And development of it can be distributed, given the open source nature, provided good programming practices and systems are maintained.

We have described its workings and architecture. But importantly we have explained several design and implementation solutions which we had to follow because of partially functioning implementations of OGC standards. We are offering anecdotal evidence of shortcomings of these standards but also of open source software. It would be worthwhile to

do a more systematic analysis but that is beyond the scope of this paper.

Acknowledgements

The ForeStClim project has received European Regional Development Funding through INTERREG IVB NEW.

References

- 52NORTH. 2011. ILWIS Community [Online]. 52North. Available: <http://52north.org/communities/ilwis> [Accessed July 2011].
- APACHE COMMONS DIGESTER. 2011. Commons Digester 3 website [Online]. Apache Commons TM. Available: <http://commons.apache.org/digester/> [Accessed July 2011].
- APACHE COMMONS HTTPCLIENT. 2011. Apache Commons HttpClient website [Online]. Apache Commons TM. Available: http://projects.apache.org/projects/commons_httpclient.html [Accessed July 2011].
- BAUD, I., PFEFFER, K., SRIDHARAN, N. & NAINAN, N. 2009. Matching deprivation mapping to urban governance in three Indian mega-cities. *Habitat International*, 33, 365–377.
- BOERBOOM, L. G. J. 2010. Integrating Spatial Planning and Decision Support System Infrastructure and Spatial Data Infrastructures Research (IJS DIR), (accepted with revision: <http://ijsdir.jrc.ec.europa.eu/index.php/ijsdir/article/viewFile/223/269>).
- BOROUGHSHAKI, S. & MALCEWSKI, J. 2010. ParcitipatoryGIS.com: A WebGIS-based collaborative multicriteria decision analysis. *Journal of the Urban and Regional Information Systems Association*, 22 23–32.
- COX, S., DAISEY, P., LAKE, R., PORTELE, C. & WHITESIDE, ARLISS 2002. OpenGIS Geography Markup Language (GML) Encoding Specification. Report no. OGC 02-023r4. Open Geospatial Consortium, Inc.
- DECISION DECK. 2011. Welcome to the Decision Deck Project website [Online]. Decision Deck Consortium, Ecole Centrale Paris. Available: <http://www.decision-deck.org/contact.html> [Accessed July 2011].
- FOERSTER, T. & STOTER, J. 2006. Establishing an OGC Web Processing Service for generalization processes. Workshop of the ICA Commission on Map Generalisation and Multiple Representation. Portland, United-states: International Cartographic Association, Commission on Map Generalisation and Multiple Representation.
- FRIIS-CHRISTENSEN, A., LUTZ, M., OSTLÄNDER, N. & LARS, B. 2007. Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions. *Transactions in GIS*, 11, 799–818.
- HE, J., CHEN, N., WANG, W. & CHEN, Z. 2009. A Dynamic Schema Matching Approach for Multi-Version Web Feature Service Retrieve.
- HERWIJNEN, M. V. 1999. Spatial Decision Support for Environmental Management, Amsterdam, Free University Amsterdam.
- KESHKAMAT, S. S., LOOIJEN, J. M. & ZUIDGEEEST, M. H. P. 2009. The formulation and evaluation of transport route planning alternatives: a spatial decision support system for the Via Baltica project, Poland. *Journal of Transport Geography*, 17, 54–64.
- MALCZEWSKI, J. 2006. GIS-based multicriteria decision analysis: a survey of the literature. *International Journal of Geographical Information Science*, 20, 703–726.
- MICHAELIS, C. D. & AMES, D. P. 2009. Evaluation and Implementation of the OGC Web Processing Service for Use in Client-Side GIS. *Geoinformatica*, 13, 109–120.
- NIJKAMP, P. 1990. Multicriteria Analysis: A Decision Support System for Sustainable Environmental Management. In: ARCHIBUGI, F., NIJKAMP, P. (ed.) *Economy and Ecology: Toward Sustainable Development*. Kluwer Academic.
- ODPM 2004. *The English Indices of Deprivation 2004 (revised)*. London: Neighbourhood Renewal Unit, Office of the Deputy Prime Minister.
- OGC 2004a. *Geolinked Data Access Service v0.9.1*. Wayland, MA: Open Geospatial Consortium.
- OGC 2004b. *Geolinking Service v0.9.1*. Wayland, MA: Open Geospatial Consortium.
- OSGI ALLIANCE 2011. *OSGi Service Platform Core Specification, Release 4, Version 4.3*. OSGi Alliance.
- PERCIVALL, G. 2010. The application of open standards to enhance the interoperability of geoscience information. *International Journal of Digital Earth*, 3, 14–30.
- SHARIFI, M. A. & BOERBOOM, L. G. J. 2006. Spatial multiple criteria decision analysis in integrated planning for public transport and land use development study in Klang valley, Malaysia. In: *Proceedings of Vol. XXXVI, part 2. Technical Commission II, ISPRS Vienna 2006 symposium, 12-16 July 2006, Austria / ed. by W. Kainz and A. Pucher. Vienna : ISPRS, 2006. pp. 125-130* This paper has been published earlier: In: *Proceedings of the ISPRS midterm conference, Commission VI, WG VI/4, Theory and concepts of spatio-temporal data chandelling and information, Vienna, Austria, 10-14 June, 2006. 7 p.*
- SHARIFI, M. A. & RETSIOS, V. 2004. Site selection for waste disposal through spatial multiple criteria decision analysis. In: *Journal of telecommunications and information technology, (2004)3, 11 p.*
- STIRLING, A. 2006. Analysis, participation and power: justification and closure in participatory multi-criteria analysis. *Land Use Policy*, 23, 95–107.
- VOLKMANN, M. 2011. *Writing API for XML (WAX)* [Online]. Object Computing, Inc. (OCI). Available: <http://java.ociweb.com/mark/programming/WAX.html> [Accessed July 2011].
- VRETANOS, P. A. 2005. *Web Feature Service Implementation Specification, Version: 1.1.0. Report No. OGC 04-094*. Open Geospatial Consortium Inc.
- ZUCCA, A., SHARIFI, A. M. & FABBRI, A. G. 2008. Application of spatial multi-criteria analysis to site selection for a local park: A case study in the Bergamo Province, Italy. *Journal of Environmental Management*, 88, 752–769.

A Vivid Relic Under Rapid Transformation

Thomas Knudsen, Simon L. Kokkendorff, Karsten E. Engsager

Abstract

TRLIB, the transformation system of the Danish National Survey and Cadastre (KMS), has served as the official tool for transformation between Danish map projections and horizontal and vertical datums for half a century. TRLIB system originated as a fairly simple system using one of the first compilers for the Algol-60 language, but it has evolved dramatically since then. Recently the extended transverse mercator implementation of TRLIB was included in PROJ, the leading open source projection package. This is well in line with other efforts towards making TRLIB more interoperable with other open source projection and transformation packages.

The main objective of this paper is to briefly outline the historical origins of TRLIB, and to describe the architectural design considerations behind an ongoing effort to make TRLIB ready for today's requirements, most notably by making it 64-bit clean and thread safe. But since the TRLIB code is used in a large number of commercial GIS products, the design considerations are also influenced by the importance of maintaining a stable, backwards compatible version, while also utilizing this golden opportunity to simplify and improve the application program interface.

Historical introduction

In the late 1950s, the Danish Geodetic Institute (Geodætisk Institut, GI) realized the need for a high performance numeric computer in order to complete a diverse range of tasks, including the adjustment of geodetic triangulation networks, adjustment of data from high precision levelling campaigns, and general support for operational mapping.

The computer, christened GIER—Geodætisk Instituts ElektronRegnemaskine (Geodetic Institute's Electronic Computer), was delivered in September 1961 (Gram et al., 1963). It was built by the Danish organization RC (RegneCentralen: Literally "the computing center", formally "The Danish Institute of Computing Machinery at the Danish Academy of Technical Sciences"). While GIER was designed in close cooperation with GI and optimized for geodetic computations, it was also remarkably successful as a generic number cruncher: a series of GIERs were built by RC and deployed in organizations around Europe, where some of them remained active for almost 20 years.

GIER was the platform for development of some of the first ALGOL60-compilers, but prior to that, the programs needed for the geodetic operations were written directly in GIERs low level assembly language.

Through a journey spanning half a century, involving a sequence of at least 4 programming languages, numerous operating systems, and an innumerable number of organizational reorganizations, these original GIER programs have evolved into what is known today as the Geodetic Software System of the National Survey and Cadastre (National Survey and Cadastre is the Danish national mapping and geodata agency, Kort & Matrikelstyrelsen – KMS).

Basically, the Geodetic Software System consists of two parts:

1. ADJ, an adjustment package, solving the complex non-linear least squares problems of geodetic network adjustment,
2. TRLIB, a 3D transformation package taking an integrated approach to coordinate transformations, providing unified access to map projections and generic reference system transformations, including horizontal and vertical datum shifts.

ADJ is primarily used for internal KMS operations, and will not be mentioned further in this paper. TRLIB, on the other hand, is widely distributed and built into a large number of commercial GIS systems, mostly in order to provide support for specific Danish map projections and datums (although TRLIB is actually a fairly generic transformation system).

The aim of this paper is to describe aspects of TRLIB, especially with respect to how its long history and ingrained 3D focus makes it differ from packages more narrowly focused on the map projection part of geodetic coordinate handling.

Approaches to numeric stability, algorithmic correctness, and evaluation of accuracy

Throughout TRLIB, care has been taken in selecting fast and numerically stable methods, e.g. Horner's scheme for evaluation of simple polynomials, and Clenshaw summation (Clenshaw, 1955, Tscherning and Poder, 1982) for evaluation of trigonometric series and other elements that can be defined as recurrence relations (Legendre polynomials et cetera).

Additional care has been taken to stand on the shoulders of giants, wherever possible. Most notably by building on the material compiled by König & Weise in their seminal reference work on the mathematical foundations of classical geodesy and cartography (König and Weise, 1951).

But selecting numerically stable and scientifically trustworthy algorithms is not enough to ensure neither numeric accuracy of results, nor correct implementation of the algorithms. But at least implementation errors lend themselves readily to analytical scrutiny and are evident once they are detected (although some pathological corner cases may be very hard to chase down).

Analytical scrutiny is much less fruitful for the other main class of "methods for obtaining wrong results": In the pioneer days of computers, subtle hardware failures were much more common than today. While not necessarily leading to catastrophic hardware breakdowns, such failures can flip the bits of the results in hard-to-detect, but exactly for that reason disproportionately catastrophic ways: once a slightly disturbed result has been entered into the geodetic databases, it may negatively influence the accuracy of network adjustments, and derived results, for years to come.

To avoid (or at least to detect) all classes of wrong results, whether induced by implementation or hardware errors, TRLIB has been designed around a policy of dual self-checking

transformations: whenever possible a forward transformation is followed by a reverse, and concluded by a check of the resulting roundtrip deviation, before the transformation result is returned. Architecturally, this can often be implemented in ways where the forward and reverse transformations to a large degree share the same code, leading to easier maintenance.

While evidently having its roots in best practice from a time where hardware stability was shaky, the method of dual self-checking algorithms remains a useful feature: For an organization as KMS that among many other roles, is also a custodian of historical geodata, it is always important to ensure that any transformations that may be applied can also be reversed. While elaborate, the dual self-checks ensures that uncaught implementation blunders and asymmetric singularities are diagnosed and reported, which is essential for long term (multiple century scale) geodata interoperability.

Datum shifts

TRLIB includes support for datum shifts. The traditional way to handle datum shifts is by the application of either 3-parameter or 7-parameter Helmert transformations. While useful (and implemented in TRLIB), the Helmert transformation does not handle tensions in the original survey networks behind the two datums transformed to and from. Traditionally such tensions are handled by applying an empirically determined transformation—either represented as a high order polynomial, or as a (NADCON style) correction grid. TRLIB supports both these approaches.

For transformation of vertical datums, things may be further convolved by the question of which geoid model was in use for assigning level coordinates to non-levelled (e.g. GPS derived) heights. TRLIB handles these complications transparently through an extensive collection of background information for each datum supported.

TRLIB also includes some support for time varying transformations based on plate tectonic motion models, for implementing the proper transformations between a global datum (e.g. ITRF89) and regional, plate fixed reference systems (such as ETRS89) at different epochs.

Coordinate systems in TRLIB which include a vertical datum are viewed as truly 3-dimensional. When performing a (horizontal) transformation between systems based on the same datum, the third/height coordinate of a point is largely irrelevant and may be viewed as an extra attribute (“2.5D”). However when doing transformations between 3D-systems based on different horizontal datums, the actual 3-dimensional position of the point is taken into consideration and the resulting horizontal coordinates depend on the third coordinate. In this sense, TRLIB is a true 3D (and not just 2.5D) transformation system.

The major challenges

While TRLIB still plays a major role as the computational work horse of the KMS, it also shows its age through artefacts from historical design decisions, that are less fortunate when running the system on modern hardware and under modern operating systems.

Currently, we are planning a restructuring of the TRLIB

code. The major problem lies in the use of global variables for the internal state, making the system inherently unsafe for execution in a parallel or multi-threaded environment. We intend to collect all internal state in an opaque object that is passed to the library functions on the stack, in a similar way to the application program interface (API) exposed by the PROJ map projection library (Evenden, 1990).

But since the TRLIB code is used in a large number of commercial GIS products, the design considerations are influenced by the importance of maintaining a stable, backwards compatible version, while also utilizing this golden opportunity to simplify and improve the API.

Comparison to PROJ

As already mentioned, we plan for a more PROJ like API for TRLIB. On the PROJ side, recently the transverse mercator implementation from TRLIB was included in the PROJ code base. In coming releases of PROJ the TRLIB code will be found as PROJ’s new *etmerc* Extended Transverse Mercator projection (Poder and Engsager, 1998, Engsager and Poder, 2007, Karney, 2011). So called because it preserves roundtrip precision far longer from the central meridian (0.03 mm at 7500 km) than the existing PROJ *tmerc*.

So while PROJ and TRLIB certainly may converge in many ways, there are still some unavoidable differences, due to PROJ’s background as a cartographic map projection library, and TRLIB’s background as a geodetic transformation library. From a user’s perspective, this becomes especially clear at the metadata level.

At the metadata level, TRLIB handles coordinate system descriptions using so called minilabels. A minilabel is a single string combining information on the projection, the horizontal datum and (if relevant) the vertical datum.

For example, the minilabel “utm32_wgs84” defines a horizontal coordinate system based on the UTM zone 32 projection of WGS84 coordinates. In PROJ, this corresponds to the definition string “proj=utm, zone=32, ellps=WGS84”.

The somewhat longer minilabel “utm32Hed50_h_dvr90” specifies a 3D coordinate system with horizontal datum being ED50 (The European Datum 1950, based on the Hayford/International ellipsoid), and vertical datum being the Danish height system DVR90.

TRLIB includes some support for translating a minilabel to ESRI-WKT format. In combination with the spatial reference system abstractions provided by the widely used *gdal/ogr* package, this provides a mechanism for automatic translation of many minilabels representing 2D-systems to PROJ parameters. For 3D-systems this is (at present) not possible since PROJ still lacks a metadata vocabulary for the description of vertical datums.

In some rare cases, TRLIB fails to produce WKT in a form that is recognizable by *gdal/ogr*. We currently consider handling this in a simplistic but robust way, by supplying a manually compiled list of minilabels and their corresponding EPSG codes.

Conclusion

We have presented the half-centennial historical background of TRLIB transformation system. Despite its age it is still vivid

and ready for the future. TRLIB was only recently released under an open source license (and made available through <https://bitbucket.org/KMS/trlib>), but in the near future we hope to implement means for better interoperability with the more well established libraries in the open source geomatics field.

Acknowledgements: We thank Willy Lehmann Weng and Knud Poder for commenting on the draft of this paper.

Bibliography

- C. W. Clenshaw. A note on the summation of chebyshev series. *Math. Tables Aids Comput.*, 9(51):118–120, 1955. URL <http://www.jstor.org/stable/2002068>.
- Karsten E. Engsager and Knud Poder. A highly accurate world wide algorithm for the transverse mercator mapping (almost). In *Proc. XXIII Intl. Cartographic Conf. (ICC2007), Moscow*, page 2.1.2, 2007.
- Gerald I. Evenden. Cartographic projection procedures for the unix environment—a user’s manual, 1990. US Geological Survey Open-File Report 90–284.
- C. Gram, O. Hestvik, H. Isakson, P.T. Jacobsen, J. Jensen, P. Naur, B.S. Petersen, and B. Svejgaard. Gier - a danish computer of medium size. *IEEE Transactions on Electronic Computers*, EC-12(5):629–650, December 1963. URL http://www.datamuseum.dk/site_dk/rc/gierdoc/ieeeartikel.pdf.
- Charles Karney. Transverse mercator with an accuracy of a few nanometers. *Journal of Geodesy*, 2011(1):1–11, 2011. doi: <http://dx.doi.org/10.1007/s00190-011-0445-3>.
- R. König and K. H. Weise. *Mathematische Grundlagen der Höheren Geodäsie und Kartographie, Erster Band*. Springer, Berlin/Göttingen/Heidelberg, 1951.
- Knud Poder and Karsten Engsager. Some conformal mappings and transformations for geodesy and topographic cartography, 1998. National Survey and Cadastre, Denmark, Publications, 4. series vol. 6.
- Carl Christian Tscherning and Knud Poder. Some geodetic applications of clenshaw summation. *Bolletino di Geodesia e Scienze Affini*, XLI(4):349–375, 1982.

Thomas Knudsen
National Survey and Cadastre
kms.dk
[thokn AT kms DOT dk](mailto:thokn@kms.dtu.dk)

Imprint - Volume 10

Editor in Chief:

Tyler Mitchell, Locate Press - info@locatepress.com

Proceedings Editor:

Gary Sherman, GeoApt LLC - gsherman@geoapt.com

FOSS4G Academic Committee:

Chair - Rafael Moreno Univ. of Colorado, Denver, USA

Thierry Badard Laval University, Canada

Maria Brovelli Politecnico di Milano campus Como, Italy

Serena Coetzee University of Pretoria, South Africa

Tyler Erickson Michigan Tech Research Institute, USA

Songnian Li Ryerson University, Canada

Jeff McKenna Geteway Geomatics, Canada

Helena Mitasova North Carolina State University, USA

Venkatesh Raghavan Osaka City University, Japan

Acknowledgements

Gary Sherman, L^AT_EX magic & layout support

Various reviewers & writers

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



This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this licence, visit: <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

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.org