# FOSS Geospatial Libraries In Scientific Workflow Environments: Experiences and Directions

Graeme McFerren[1], Terence van Zyl[1], Anwar Vahed[1]

August 2, 2010

[1]ICT4EO Research Group, Meraka Institute, CSIR, Pretoria, South Africa

### Abstract

In numerous research fields, scientists are increasingly turning to e-science and specifically scientific workflows as a way of improving, broadening and hastening their results. Enhanced collaboration, on-demand access to tools, data and high performance processing facilities are some of the gains to be made. Scientific workflows are concerned with, amongst others, supporting the repeatability and provenance of experiments. Scientific workflows have had a measure of success in the astronomy, bio-informatics, chem-informatics, geophysics and eco-informatics domains [Gil, et. al., 2007]. This paper describes our initial investigations into developing geospatial Scientific Workflows to support researchers in exploring, integrating and visualising Earth Observation and GIS data in conjunction with other research data. We describe some of the functionalities we require in the context of three sets of research endeavour - wildfire research, flood modelling and the linking of disease outbreaks to multi-scale environmental conditions. We note the relative lack of support for geospatial data, services and functions within some of the FOSS Scientific Workflow packages. This paper highlights challenges and results of utilising various FOSS geospatial libraries within these Scientific Workflow environments.

# 1 Introduction

Scientists faced with growing volumes of data, and the need to perform complex calculations on these data in distributed and collaborative environments, have turned to the concept of scientific workflows. Research indicates that scientists can be shielded from the underlying technologies and some of the processes needed to analyse these data sources through the mechanism of scientific workflows, which automate complex processes and provide integrated access to datasets that are often characterised by their large sizes and distributed locations [Gil, et. al., 2007, Gibson, et. al. 2007, B. Ludascher, et. al. 2006]. Scientific workflows are aimed at formalising and automating the various analytical, transformative and visualisation steps in a data flow, while also supporting reproducibility of experiments and knowledge sharing. Scientific workflows have the potential to reshape the scope of the scientific analysis process and to help researchers to continue to perform sound science at a greater pace.

This paper introduces several scenarios to support the idea of *geospatially enabled scientific workflows*. To an extent, this concept is addressed by well-known tools such as ModelBuilder from ESRI [ESRI, 2006] or SEXTANTE [SEXTANTE, 2010] in the FOSS4G world. These tools however, are firmly focused on the GIS space, whilst many scientists are simply using GIS as an adjunct to their investigations, implying the need to introduce geospatial functionality into generic scientific workflow environments such as Kepler [Altintas, et. al., 2004].

We believe that FOSS4G tools offer many possibilities for geospatially enabling scientific workflows, and allow great possibilities for experimentation, which is at the heart of the scientific workflow process. The array of FOSS4G tools available means that many tasks can be accomplished through the low cost and great flexibility inherent in these tools. Our initial experiments with these scenarios revealed some technical difficulties, however, which are elucidated in this paper. These difficulties lead us to a few ideas on how to proceed with geospatially enabling scientific workflows environments.

The following section presents the scenarios that served as test bench for exploring scientific workflows that utilise some FOSS4G tools for the given scenarios. Section 3 describes our findings, and the concluding section presents some recommendations that would ease the integration of geospatial tools in scientific workflows.

# 2 Scenarios

In the following sections we detail some scenarios in which we envisage geospatially enabled scientific workflows could play a strong supportive role. The scenario for flood risk simulation was used to explore the domain and test initial ideas, such that future work on the other scenarios is more informed.

## 2.1 Wildfire research

The CSIR has for a number of years been active in wildfire research in southern Africa, considering the phenomenom from multiple angles. Scientists from the organisation have been utilising earth observation data to investigate wildfire occurence, patterns and structure. The South African Advanced Fire Information System (AFIS) is a well documented application that consists of a moderately large and growing database of active fire detections, burned areas, weather records and infrastructure vulnerable to wildfire [McFerren and Frost, 2009], [AFIS Web, 2010]. Coupled to this database is an alerting system that warns of wildfire threats to infrastructure.

The exposure of these AFIS datasets to scientific workflow environments is desirable in order to support or enable various research efforts, some of which are detailed below:

### 2.1.1 Mega-fire Analysis

Researchers are interested in determining, for various land cover types, what the characteristics of so-called mega-fires are. Mega-fires are a function of the size, intensity, duration and impact of wildfires. This would be an investigative scientific worklow that would need to be re-executed periodically on new data, perhaps with changed parameters (new landcover scheme or temporal overlap variable, for example).

The initial analysis run was undertaken in a manual process utilising GRASS 6.4 [GRASS, 2008], GDAL 1.6 [GDAL, 2010] and PostGIS 1.4 [PostGIS, 2010] to process a combination of ten years of MODIS burned area data with ten years of MODIS active fire data and large landcover datasets. The aim was to merge the burned areas into spatially and temporally contiguous clumps, essentially fires, and assign fire radiative power values from spatio-temporally coincident active fires. The process would then extract fires of the highest intensity, longest duration and biggest size, essentially the mega-fires.

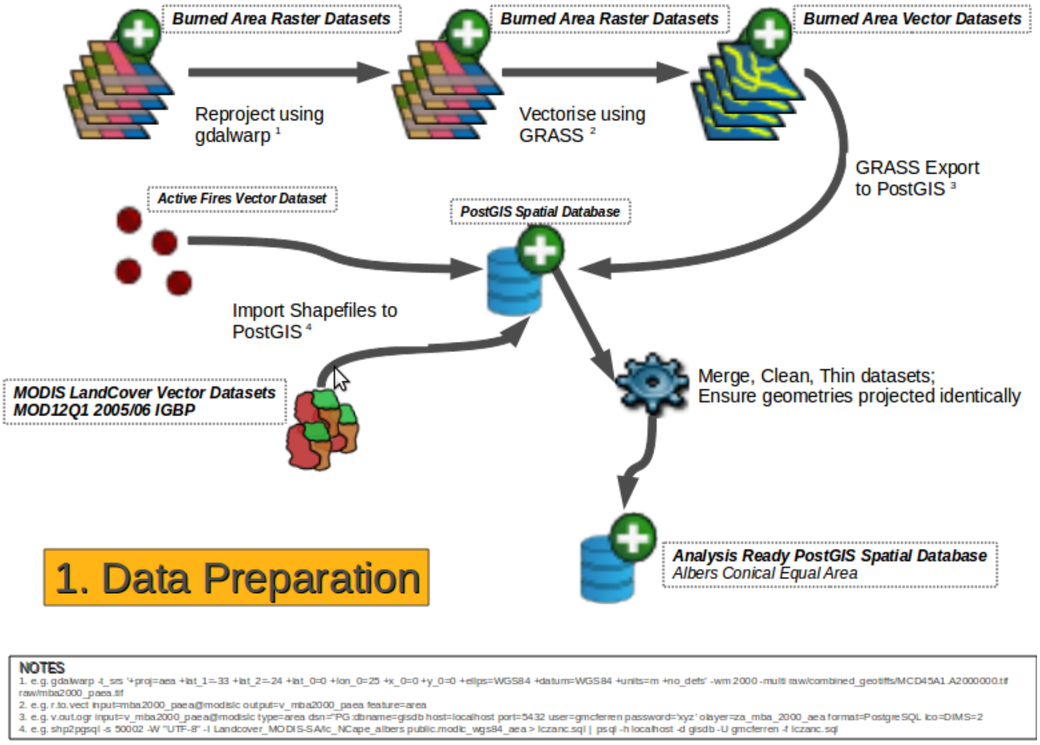
Table 1 enumerates the kinds of functionality utilised in the manual workflow, and the software tools that provided the functionality, while Figure 1 illustrates the data preparation workflow for this scenario. The manual process to generate mega-fires completed in roughly 3 days of elapsed time. Ideally, a scientific workflow environment would allow this process to be 'fire-and-forget'.

| **Process Requirement** | **Software Tool** |
|---|---|
| Ingest vector data from shapefiles, projecting from/to custom projection | PostGIS shp2pgsql |
| Simplify and clean certain geometries | PostGIS ST_Dump, ST_Buffer |
| Indexing vector data | PostGIS GiST R-Tree |
| Reprojection of raster data (e.g. MODIS burned areas) | gdalwarp |
| Ingestion of rasters to GRASS | GRASS r.gdal.in |
| Vectorisation of burned areas rasters | GRASS r.to.vect |
| Export to spatial database | GRASS v.out.ogr |
| Data thinning, attribution, merging | PostGIS SQL select/delete, update, insert queries |
| Geometry Unioning on Intersection | PostGIS ST_Intersects, ST_Union |
| Data merging | PostGIS insert SQL |
| Mixed Spatial/Attribute Joins | PostGIS select SQL, ST_Intersects |
| Compute temporal adjacency | PostGIS Custom Function over date attributes |
| Spatial overlay to extract fire intensity statistics from spatially and temporally co-incident active fires | PostGIS SQL utilising statistical functions, ST_Intersects, Postgres temporal operators - Overlaps and windowing functionality |
| Cookie-cut areas of interest | PostGIS ST_Within |
| | |

Table 1: Mega-fire scientific workflow functional requirements and supporting software

Four main requirements of a geospatially enabled scientific workflow environment can be identified for this scenario:

1. Access input data
2. Ingest data into various environments, in this case PostGIS and GRASS
3. Run arbitrary command line processes against spatial processing/transformation engines or issue arbitrary, complex SQL against a spatial database
4. Access results and pass into next step of process


Burned Area Raster Datasets
Reproject using gdalwarp [1]
Burned Area Raster Datasets
Vectorise using GRASS [2]
Burned Area Vector Datasets
GRASS Export to PostGIS [3]
Active Fires Vector Dataset
PostGIS Spatial Database
Import Shapefiles to PostGIS [4]
MODIS LandCover Vector Datasets
MOD12Q1 2005/06 IGBP
Merge, Clean, Thin datasets;
Ensure geometries projected identically
1. Data Preparation
Analysis Ready PostGIS Spatial Database
Albers Conical Equal Area
NOTES

```
1. e.g. gdalwarp -t_srs '+proj=aea +lat_1=-33 +lat_2=-24 +lat_0=0 +lon_0=25 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs' -wm 2000 -multi raw/combined_geotiffs/MCD45A1.A2000000.tif raw/mba2000_paea.tif
2. e.g. r.to.vect input=mba2000_paea@modislc output=v_mba2000_paea feature=area
3. e.g. v.out.ogr input=v_mba2000_paea@modislc type=area dsn="PG:dbname=gisdb host=localhost port=5432 user=gmcferren password='xyz' olayer=za_mba_2000_aea format=PostgreSQL lco=DIMS=2
4. e.g. shp2pgsql -s 50002 -W "UTF-8" -I Landcover_MODIS-SA/lc_NCape_albers public.modlc_wgs84_aea > lczanc.sql | psql -h localhost -d gisdb -U gmcferren -f lczanc.sql
```

Figure 1: Data preparation workflow for mega-fire analysis

The number of commands available to a user by the combination of GDAL, GRASS and PostGIS is reasonably high, reaching into the hundreds, perhaps thousands. This is one of the chief challenges facing would-be implementors of geospatially enabled scientific workflow environments: where does one start?

### 2.1.2 Area burnt per area, per time period

This would be a simpler investigative scientific worklow aimed at providing statistics for an arbitrary area concerning the extent of burnt areas within it, perhaps for a given time period. This workflow could be more readily exposed via standardised interfaces, such as an Open Geospatial Consortium (OGC) [OGC, 2010] Web Processing Service (WPS), since the process itself is more structured and atomic than the mega-fires analysis described above. Indeed, the inputs for this service could be drawn in from OGC Web Feature Services (WFS) . Outputs would typically be in CSV text files, perhaps for input into other scientific worklows concerned with estimating gas and particulate matter emissions for a municipality/ county.

### 2.1.3 Support for detailed wildfire alerts

This kind of scientific workflow would be more akin to a business process workflow, but would differ in taking advantage of scientific computing resources to process alert variables and generate visualisations. In this scenario, managers of infrastructure potentially vulnerable to wildfire, would receive alerts of active fires, fire weather and other data. Such alerts would be transmitted in a 'tactical alert cube' where a context to the alert would be carried, perhaps as a small visualisation of the fire on a terrain map, with data about vegetation condition, burnt areas in the vicinity and nearness to access points, for example. The basic geospatial analysis and processing needs can clearly be seen, but when you have 500 fires and thousands of potential alertees, the issue becomes computationally more intensive, increasing the need for more powerful computational capabilities. Within this scenario, then, there is not much exploratory or scientific work, but the processing requirements may be enough to necessitate the use of e-science infrastructure. In addition, alerts such as these may be triggers or inputs for other wildfire related scientific workflows, and can also be the payload for alerting protocols.

## 2.2 Links between environmental conditions and disease outbreak

CSIR Natural scientists (primarily microbiologists and environmental modelers) are interested in establishing whether theories and hypotheses about biophysical processes associated with vibrio cholerae ecology and cholera disease dynamics developed for, in particular, Bangladesh [Lipp, et. al., 2002], are applicable in the southern African region. This cholera scenario is of interest for its complex data and processing requirements (whether remote sensed, in-situ or contextual in origin), environmental modelling component and importance in terms of human health in areas of southern Africa. Scientific workflows may offer natural scientists a potential simplifying mechanism for using the facilities of the sensor web to answer spatio-temporal 'window' queries such as "did rainfall in area x, over period y, have any influence on variable z" [McFerren, et.al. 2008]. For such researchers, geospatially enabled scientific workflows would assist in:

- discovery of and access to the relevant slices of data
- abstracting access to heterogenous earth observation datasets and transformation of such data into formats or encodings that are compatible with particular scientific tools
- tracing data and processing lineage
- validating or ground truthing their primary datasets or even their models with measurements and observations
- alerting researchers to noteworthy changes in the environment
- guiding researchers through the sometimes complex tasks of working with spatial data
- rapid and simple orientation to datasets, services and resources before they get used - what is this dataset intended to be used for? what types of fields are available in the dataset? does this dataset fall within a specific area of interest? who produced this dataset? am I allowed to use it?

Ultimately, the researchers would be performing interactive science, focused on data, not formats or underlying software tools.

## 2.3 Flood simulation

We investigated flood simulation as a contrived academic exercise in exploring some of the capabilities of the Kepler scientific workflow environment - in particular its ability to ingest and process data from OGC standards compliant web service interfaces. The notion was to combine rainfall data from satellite remote sensing (TRMM), data from river gauging stations, a Digital Elevation Model and spatial data of informal settlements, in order to estimate which communities would be vulnerable to flooding after severe storms. We employed no detailed hydrological model, rather, we simply accessed data from the web services and trivially processed the data, allowing us insight into the difficulty of incorporating such services into scientific workflow environments. Figure 2 shows an example of a workflow constructed in Kepler using some of these functionalities.

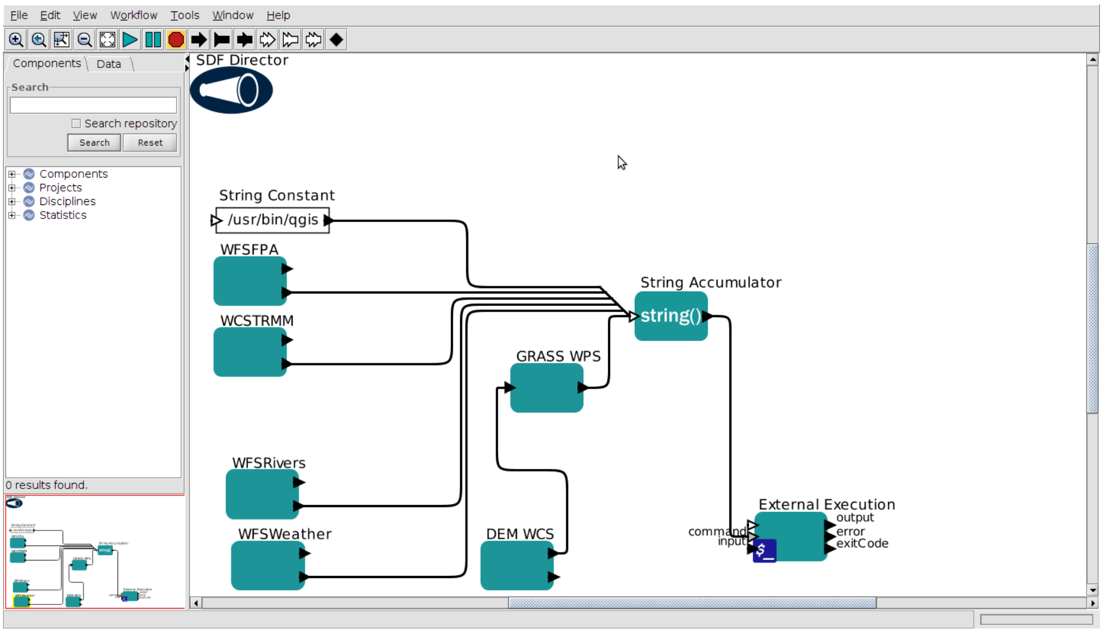


Figure 2: Example flood simulation workflow.

The web service standard interfaces under investigation included OGC Web Feature Service (WMS), Web Map Service (WMS), Web Processing Service (WPS), Sensor Observation Service (SOS), and Web Coverage Service (WCS). The data and processes behind these services were presented in the following ways:

- TRMM data in geotiff format was presented through a WCS
- DEM data presented as a geotiff through a WCS
- river gauge data presented as Observations & Measurements through a SOS
- municipal demarcations were presented as GML through WFS
- an elementary spatial overlay, performed in GRASS, was placed behind a WPS

Finally, when completed, the workflow instantiated QGIS with mapped results visualised.

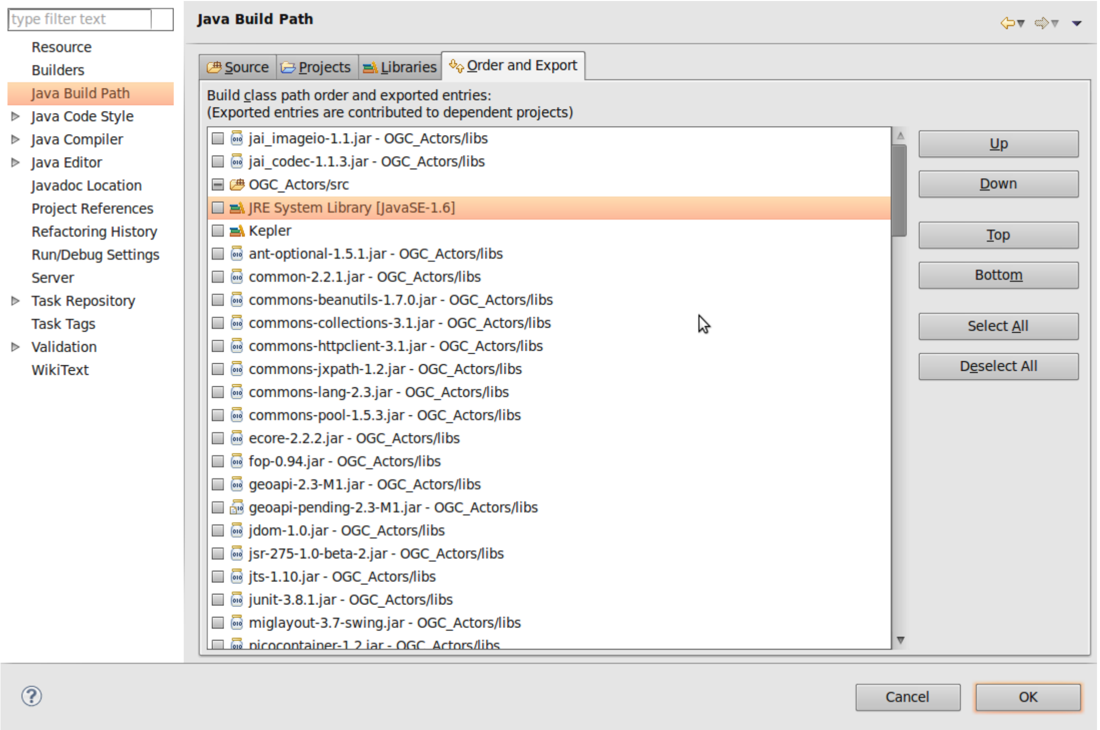
## 3 Experiences

| Workflow environment | Primary language | FOSS4G components utilised | Evaluation level |
|---|---|---|---|
| Kepler | Java | GeoTools | Extensive |
| | | 52North OX-Framework | Extensive |
| | | QGIS | Extensive |
| | | GRASS | Extensive |
| | | GDAL | Extensive |
| | | uDIG | Precursory |
| | | ows-framework | Precursory |
| VisTrails | Python | PostGIS | Precursory |
| | | owslib | Precursory |

Table 2: Workflow environments, libraries, tools and components evaluated

Table 2 presents a list of the libraries, tools and components that were evaluated in the context of the scenarios in section 2. The research method followed was a case based study. The main intention was to rapidly build, using off-the-shelf components, a semi-operational workflow system supporting the flood risk simulation scenario, with the intention being to carry our learning through to other scenarios outlined in section 2. The overall result was not as successful as originally expected. Here we outline some of the issues that hindered the rapid application development approach when attempting to perform various raster and feature transformations, build clients for WFS, WCS, WMS, SOS and WPS and visualise results:

- No single package appears to cover all or even most of the required open standards, forcing developers to integrate a number of different libraries into a workflow environment. Each package carries its own design styles, internal data models and set of supported functionalities.
- Issues with dependancy conflicts between versions of libraries exist. It is not always possible to draw binary libraries (geospatial or otherwise) into workflow environments, and certain components have explicit dependencies that conflict with current versions of other components, sometimes making it necessary to resort to compiling components with older or untested library dependencies. Figure 3 shows a small view of some of these libraries used in the Kepler workflow environment to develop actors for accessing OGC services.
- Various libraries have different patterns of usage with some requiring extensive coding to set up.
- Differing internal data models between libraries makes it difficult to compose workflows with components and actors derived from different libraries. Even when the data models were similar, it required going beyond the standard interfaces to get at this functionality.
- Visualisation was particularly difficult, exemplified by the well known problem of coordinate axes varying between different implementations of the WFS or different versions of the standard. Eventually in order to overcome these inconsistencies all feature data was exported to shape files and visualised in QGis.



Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

- □ jai_imageio-1.1.jar - OGC_Actors/libs
- □ jai_codec-1.1.3.jar - OGC_Actors/libs
- □ OGC_Actors/src
- □ JRE System Library [JavaSE-1.6]
- □ Kepler
- □ ant-optional-1.5.1.jar - OGC_Actors/libs
- □ common-2.2.1.jar - OGC_Actors/libs
- □ commons-beanutils-1.7.0.jar - OGC_Actors/libs
- □ commons-collections-3.1.jar - OGC_Actors/libs
- □ commons-httpclient-3.1.jar - OGC_Actors/libs
- □ commons-jxpath-1.2.jar - OGC_Actors/libs
- □ commons-lang-2.3.jar - OGC_Actors/libs
- □ commons-pool-1.5.3.jar - OGC_Actors/libs
- □ ecore-2.2.2.jar - OGC_Actors/libs
- □ fop-0.94.jar - OGC_Actors/libs
- □ geoapi-2.3-M1.jar - OGC_Actors/libs
- □ geoapi-pending-2.3-M1.jar - OGC_Actors/libs
- □ jdom-1.0.jar - OGC_Actors/libs
- □ jsr-275-1.0-beta-2.jar - OGC_Actors/libs
- □ jts-1.10.jar - OGC_Actors/libs
- □ junit-3.8.1.jar - OGC_Actors/libs
- □ miglayout-3.7-swing.jar - OGC_Actors/libs
- □ picocontainer-1.2.jar - OGC_Actors/libs

Figure 3: Snapshot of some Kepler OGC Actor dependencies.

- Java is an inherently structured language, which often requires a large amount of 'plumbing', linking and compiling to use in a meaningful way. In the context of the case studies this structure sometimes became a hindrance, for fluidity and constant refinement is a common characteristic of scientific workflows especially where scripting and hacks would have sufficed.

Despite these difficulties, there were some aspects that worked well:

- Command line tools can be readily used from scientific workflow engines. Such tools have clearly defined sets of inputs and outputs along with clearly defined parameters. Kepler, for example, provided good support for parameterising and executing command line tools. This opens up the possibility for wide usage of powerful tools like GRASS, with some pre-scripting and setup effort.

- Shapefiles seem to be better supported than GML and often provided a more reliable interchange format.

- QGIS provided a very useful end point for visualisation. However, it was not immediately apparent how visualisations within the QGIS environment could be drawn back into the scientific workflow environment.

# 4 Directions

Based upon the scenarios which we explored and the experiences we had in implementing a scenario, here we present some recommendations for improving the integration of geospatial tools into generic scientific workflow environments:

- It would be ideal if libraries exported a common API for working with their geospatial objects (or, use API's consistently). There is great re-use of particular foundational geospatial libraries (e.g. JTS/GEOS) amongst the components we investigated, but it is difficult to work with the components generically. This is exemplified by the difficulties in passing geospatial objects around scientific workflow environments, with the result that serialisation is often required.

- In turn, we experienced mixed support for interchange formats such as OGC GML and its derivatives. Once again, consistent support for this standard amongst the various geospatial components would be beneficial.

- Libraries that support command line interfaces tend to be more readily used in scientific workflow environments. This allows workflow developers to focus on scripting of commands rather than having to program against a potentially complex API.

- The use of standardised web service interfaces is a powerful mechanism to reduce some of these problems, at the cost of some inefficiency. The scientific workflow environment can be freed from having to deal with dependencies and library version conflicts, for example, and become more of a coordinator of messages being passed between services. An implication is the need for libraries that expose complete client implementations for most or all the major OGC standards. In the Python world [OWSLib, 2010] is attempting to accomplish this task, whilst in the Java environment, we came across [ows-service-framework, 2010].

- Visualisation of spatial data in various ways is important in scientific workflows - users are not just interested in transformations and processing, but in seeing how results look, sometimes even as the workflow is running. An interesting example would be to expose a number of lightweight mapping canvasses into a scientific workflow component that implements a spreadsheet metaphor. This would allow time-series visualisation, perhaps the ability to zoom the focus of a running scientific workflow in on a certain area as examples of things that could be accomplished in this way.

- We believe that bringing spatial-tomporal data capability into scientific workflows allows for interesting kinds of workflow patterns. One example is the idea of "listener workflows" which automatically respond, perhaps by executing sub-workflows, taking a sample of a sensor data stream perhaps, when certain events happen at certain places.

It is our view that integrating geospatial capabilities into scientific workflow environments is an area that needs much attention, with the opportunity to help non-GI scientists harness the wide variety of rich FOSS4G componentry better to make sense of their data.

# References

[Altintas, et. al., 2004] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, S. Mock. Kepler: an extensible system for design and execution of scientific workflows. *Scientific and Statistical Database Management*, 2004. Proceedings. 16th International Conference on In Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on (23 June 2004), pp. 423-424

[Gil, et. al., 2007] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau and J. Myers, “Examining the Challenges of Scientific Workflows”. IEEE Computer, vol. 40, no. 12, IEEE Computer Society, pp. 2432, December, 2007

[AFIS Web, 2010] Advanced Fire Information System. http://afis.meraka.org.za/ . Retrieved 20-07-2010

[McFerren and Frost, 2009] G. McFerren and P. Frost., 2009. The Southern African Advanced Fire Information System. Information Systems for Crisis Response and Management (ISCRAM 09) Conference, Göteborg, Sweden, 10 May 2009, Pages: 6

[McFerren, et.al. 2008] G. McFerren, T. van Zyl, M. van der Merwe, M. du Preez, 2008. User Requirements for Sensor Web based Scientific Workflows in the Cholera Research Domain. IGARSS (International Geoscience and Remote Sensing Symposium) 2008 Proceedings, Boston, USA

[Gibson, et. al. 2007] A. Gibson, M. Gamble, K. Wolstencroft, T. Oinn and C. Goble, ’”The Data Playground: An Intuitive Workflow Specification Environment”. Proceedings of the Third IEEE International Conference on eScience and Grid Computing. IEEE , 2007 [7]

[B. Ludascher, et. al. 2006] B. Ludascher, S. Bowers, T. McPhillips and N. Podhorszki, “Scientific Workflows: More eScience Mileage from Cyberinfrastructure”, Proceedings of the Second IEEE International Conference on eScience and Grid Computing (e Science’06), IEEE , 2006

[SEXTANTE, 2010] SEXTANTE, www.sextantegis.com. Accessed 23-07-2010.

[ESRI, 2006] Environmental Systems Research Institute, Inc. An overview of ModelBuilder. http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=An_overview_of_ModelBuil Accessed 23-07-2010.

[GRASS, 2008] GRASS Development Team, 2008. Geographic Resources Analysis Support System (GRASS) Software. Open Source Geospatial Foundation Project. http://grass.osgeo.org

[GDAL, 2010] GDAL. 2010. GDAL - Geospatial Data Abstraction Library: Version 1.6.3., Open Source Geospatial Foundation, http://gdal.osgeo.org. Accessed 23-07-2010.

[PostGIS, 2010] PostGIS. 2010. PostGIS - Support for geographic objects to the PostgreSQL object-relational database: Version 1.4.1., Refractions Research, http://postgis.refractions.net. Accessed 23-07-2010.

[OGC, 2010] OGC Standards and Specifications. http://www.opengeospatial.org/standards. Accessed 23-07-2010

[OWSLib, 2010] OWSLib. http://trac.gispython.org/lab/wiki/OwsLib. Accessed 23-07-2010.

[ows-service-framework, 2010] ows-service-framework. http://code.google.com/p/ows-service-framework/. Accessed 23-07-2010

[Lipp, et. al., 2002] E.K. Lipp, A. Huq and R.R. Colwell, 2002. Effects of Global Climate on Infectious Disease: the Cholera Model. Clinical Microbiology Reviews, Vol. 15, No. 4, p. 757-770. American Society for Microbiology