Beyond GIS with EO4V is Trails: a geospatio-temporal scientific workflow environment

by Terence van Zyl, CSIR

Abstract

The paper presents the problems associated with current geographic information system (GIS) and the need for their integration with other specialised tooling such as statistics in R or matrix algebra of Octave. The authors take the stance that neither GIS should be adapted to include all the functionality of for example a statistics package nor should the statistics package be adapted to become a full GIS. Instead by utilising the advantages of scientific workflows many paradigms can be accommodated at once. The scientific workflows approach has other advantages to such as provenance, repeatability and collaboration. The paper presents EO4VisTrails as an example of such a scientific workflows approach to integration and discusses the benefit of the use of standards in combination with this approach taken in light of two projects that have been completed.

Keywords

scientific workflows, EO4Vistrails, GIS, cloud computing

Introduction

In previous work we explored the concept of geospatially enabled scientific workow environments where geospatial tools, functionalities and data are used in conjunction with other research tools, functionalities and data, from, for example, the visualisation, numerical modelling and simulation, computational intelligence, high performance computing and statistical domains [4]. The culmination of this effort has resulted in the imminent release of the first stable release of EO4VisTrails (https://code.google.com/p/eo4vistrails/). EO4VisTrails is an extension to the scientific workflow, visualisation and data provenance system VisTrails (http://vistrails.org/) that provides functionality for working with geospatial libraries, data stores, data types and services. This paper will look at EO4Vistrails and the concepts of geospatially enabled scientific workflows as it is applied in real world applications, distilling out the lessons learnt and presenting an argument for the effectiveness of the same technology applied in other similar and related projects [1].

In order to appreciate EO4Vistrails one needs to first understand the functionality given by Vistrails as a scientific workflow environment. The key aspects provided by Vistrails are a visual programming paradigm combined with the ability to compose chains of interrelated data manipulation and scientific processing steps into a single re-executable, transportable and compose-able entity. This entity takes the form of a self describing file that can be shared amongst scientists. In addition the file stores any comments and meta data along with the full provenance of that given scientific process [2].

A defining characteristics of earth observation science is the nature of the data used during the scientific process. Typically the data is acquired by observing the spatial and or temporal unfolding of some phenomenon. As a result of the spatial component of the data there has been a rapid adoption of geographic information systems (GIS) as a way of analysing this data. GIS together with the tools presented by geostatistics have formed the foundational layer of much of earth observation science. In recent times the temporal component of the data has become more prominent in earth observation science. This prominence is characterised by a more recent focus on temporal analysis of spatial data. As a result the use of GIS in its current state and its functionality needs to be reviewed.

EO4VisTrails recognises several important emerging technical and user requirements surrounding earth observation and geospatial data [1, 3, 4, 5]:

- That process automation and the repeatability of processing is crucial to the realisation of good science and product innovation.
- The need for exploration in which a GIS practitioner is able to evaluate how various parametrisations effect the outcome of a model.
- The time dimension and third spatial dimension is increasingly important to researchers working with geospatial data.
- The need for information communication technology (ICT) resource management in the form of access to cloud computing, high performance computing and very large databases are critical components of research [6].
- GIS tools are not necessarily the key tools used by researchers in their spatio-temporal analytic work – other platforms such as R or Octave are just as valuable and need to be integrated.
- Earth observation and geospatial data are often massively large, requiring out-of-core computations.

- Data and processing capabilities are often found behind complex web service interfaces.
- It is not always advisable to transfer earth observation and geospatial data between platforms, sometimes it is more pragmatic to move the analysis/processing code instead.
- Researchers wish to focus on their research, not on web service protocols or geospatial data transformations [3].
- The need to bridge the gap between scientific research and product innovation and development.

To support these points, EO4VisTrails provides web service aware components, distributed and multiprocess aware helper utilities inherited from RPyC4Vistrails (http://code.google.com/p/rpyc4vistrails/), and access to scientific data types and libraries. The view of earth observation (EO) or geospatial data drops in importance somewhat, with the focus on shaping such data for use in numerical modelling, simulation and statistical tools in addition to the primary geoinformation aware tools. EO4VisTrails attempts to provide transparent geospatial data transformations, where possible between each of these paradigms.

VisTrails itself is a PyQT (http://www.riverbankcomputing.co.uk/software/pyqt/intro) based application. This has two notable effects. Firstly, modules and extensions of VisTrails are built in Python and can harness much of the large array of scientific computing libraries that expose a Python application program interface (API). In fact, VisTrails includes libraries such as Matplotlib (http://matplotlib.sourceforge.net/) and VTK (http://www.vtk.org/) in its base configuration, while user-space packages for NumPy and SciPy are also provided (http://numpy.scipy.org/). Secondly, VisTrails utilises the graphical user interface and non-gui functionality provided by the rich, underlying QT set of libraries – this assists in rapid development of any GUI components necessary as a Vistrails extension. This use of QT allows for ready mappings between VisTrails and the QT/PyQT based QuantumGIS QGIS (http://www.qgis.org/).

Vistrails is a desktop-based application, but uses Python mechanisms for accessing Internet based data and services. VisTrails' core, and extensions to it, consist of packages of modules. Functionality is contained in these modules; scientific workflows are constructed by pipe-lining various modules. Modules are connected via strongly typed input and output ports. Modules are not linked until runtime, but there is opportunity at configuration time to parametrise modules and inspect meta-data about potential module input.

This paper describes the broad design decisions of the chosen platform, introducing the EO4VisTrails extension, before describing the various geospatial and high performance computing components that it delivers. We then briefly discuss some implementation experiences, before addressing some challenges we experienced and future research directions. In Section 2 some background to scientific workflows is presented. In Section 3 the design of EO4Vistrails is explore. Some applications of EO4VisTrails are explored in Section 4 and we give our results and lessons learnt. Section 5 closes with some conclusions.

Scientific workflows

Extensive comparisons have been made between business workflows and scientific workflows and a clear case for the later as an alternative and “self existent” technology has been made. Below the authors outline some of the most salient differences as these will help in examining the function of geospatio-temporal scientific workflows and how these domain specific scientific workflows differentiate themselves. In short the field of scientific workflows examines the use of business workflows as a technology within the scientific domain. Applying the technologies related to business workflows to the science domain comes with some unique constraints, these are discussed below [5, 11].

First, workflows in the science domain require additional support for process exploration and design as opposed to just representing a given business process. This process exploration needs to include data exploration and allow the applicant to move back and forward in time looking at previous workflow designs and following old leads.

Secondly, the end result of workflow exploration and design is as important as how the applicant got there. Thus the provenance relating to the design process needs to be captured along with the end result. Business workflows are concerned with the end result only.

Third, scientific workflows must support parameter exploration in which the applicant can evaluate the same workflow on the same data but given different parametrisations.

Repeatability is also an important aspects in scientific workflows where one is able to follow the exact same steps using the same input data and again arrive at the same result. However, a business is unlikely to repeatedly perform the same workflow on the same customer to see if the outcome is repeatable.

Another key component of scientific workflows is collaboration in the form of sharing between individuals and organisations, this is in contrast with a business workflow which will seldom be shared. As a result standards are essential for interoperability, sharing and reuse in scientific workflows [1, 7].

Finally, scientific workflows are strongly data-oriented; each step tends to replicate and then transform the data from the previous step [11]. Business workflows tend to be process-oriented where each step captures some business process.

EO4VisTrails design

Limitations of current approaches

The geospatial domain has a long history of using ICT in assisting in driving forward science. This history and technology takes the form of what is termed "classical GIS". Although this provides a solid foundation to work from it also presents some major challenges with respect to the uptake of a new technology paradigm.

Increasingly geospatial scientists are needing to turn to tools outside of their traditional GIS tools and platforms to perform complex computational operations. These tools are many; some of which take the form of scripting languages with an emphasis on either statistics or algebra, but which are not particularly suited to the visual component of geospatial data analysis, nor to the unique requirements of dealing with data with an explicit spatial component.

On the other hand, traditional GIS tools are increasingly required to provide additional statistical and algebraic functionality to meet their users requirements for more complex computations. Additionally the introduction of time and additional spatial dimensions has left these tools with many gaps to fill.

Neither of these solutions, the extending of statistical or algebraic scripting environment or the extending of GIS address any of the new user requirements described in Section 1 relating to provenance, repeatability and collaboration. The approach expounded here by the authors is that of the classic UNIX philosophy. That is, for each tool to "do one thing and do it well", with a coherent system or workflow being created through the use of piping mechanism to join these modules together.

UNIX Philosophy in EO4Vistrails

EO4VisTrails takes exactly the UNIX approach when dealing with the challenges of addressing GIS needs, specialist scripting languages and additional functionality such as network analysis or data preprocessing. Instead of trying to build one monolithic application EO4VisTrails provides the framework in which various GIS, statistical and algebraic scripting, can be brought together to form a single coherent geospatio-temporal scientific workflow while providing the applicant with the existing Vistrails' functionalities of performing both data and process exploration, repeatability, provenance, automation and collaboration.

VisTrails is written in Python and as such provides the precedent for the preferred language of other tooling within EO4VisTrails. As such the Python philosophy of "don't reinvent but wrap and adapt" is taken to be part of the design - VisTrails provides a framework in which new modules can be created - either directly via an in-built "PythonSource", or by creating your own "wrapper" modules and then providing these via a package [8].

EO4VisTrails makes use of the GIS capabilities of QGIS, which does one thing and does it well, together with the scripting and analytical capabilities of various scripting languages including, Python, R (http://www.r-project.org/) and Octave (http://www.gnu.org/software/octave/) each of which are specialist languages in and of themselves. It naturally takes advantage of the existing collaborative, provenance and workflow capabilities of VisTrails and provides a cohesive framework in which geo-processing chains can not only be presented but also explored. In the previous section the authors presented the high level architecture of EO4VisTrails, in the following section some examples of its use will be presented [8].

The core framework providing the automated workflows, provenance, collaboration and repeatability aspects to EO4VisTrails is the scientific workflows and visualisation application framework of VisTrails. Additionally EO4VisTrails provides an integration framework between various programming languages, data formats and standards within the EO domain. Within this framework, some mechanisms enabling integration are required. In this regard, EO4VisTrails makes extensive use of Numpy and GML. Both paradigms provide various in memory and on-disk mechanisms for moving data in and out of various frameworks.

Applications and lessons learnt

Currently the EO4VisTrails package has been used in a number of successful projects, hinting not only at the ability to tackle many of the user requirements described in section , but also at the potential for much wider application. Here the authors outline two of these projects and describe how the use of EO4VisTrails has made the project more successful. Additionally, we present the outcomes of these projects and some lessons learnt. We also present some additional projects that currently, or in future, will make use of EO4VisTrails.

EO4HiTempo

Description

The first system we describe is the use of EO4HiTempo to perform time series analysis. The EO4HiTempo project required a significant amount of CPU processing of a massively-parallel change detection algorithm, Mann-Kendall, on a 10 year timeseries of four Moderate Resolution Imaging Spectroradiometer (MODIS) tiles. The functional description of this project is:

- The input to the algorithm comes from a non-standardised HDF-5 cube as an output of the HiTempo project [9].
- the Mann-Kendall algorithm is not available off the shelf as a standard Python package, but is available in R. However, R is resource hungry and unable handle the full data set in a reasonable (30 TB) amount of RAM.
- The results need to be chained together and correct geographic meta-data added so as to make the result GIS ready, this is done using GDAL from python.
- The entire process is resource hungry requiring far more RAM and CPU than is available on a desktop.

As a result of the nature of this project it has some interesting non-functional requirements:

- The need to effciently perform the same operation many times, in this case a (2400 x 2400 x 4) times series, but where each operation is independent of the others.
- The need to integrate heterogeneous tools and languages.
- The need for some type of distributed or cloud-based approach to handle the data and processing size [10].

Results and lessons learnt

The EO4HiTempo project was completed successfully. The project demonstrated scientific workflows ability to provide scientific exploration of process and of data. In addition the rich provenance allowed for repeat experimentation using various parametrisations. The integration of GIS and statistical processing packages with each performing its own specialised task well also showed the utility of this approach. In addition the massive speedup obtained as a result of the use of high performance cloud computing resources in a distributed fashion made possible what would be an unobtainable result on a desktop. The process was able to complete in a about four (4) hours compared to a desktop result that would have taken many days.

In summary the scientific workflow approach to integrated GIS with specialised statistical packages using high performance cloud computing within EO4HiTempo project resulted in effective data and process exploration, parametrisation and integration of high performance cloud computing resources. This can be seen in the fact that originally R was unable to load a full MODIS tile and process it. Even when the tile was broken into smaller tiles the manual process was prohibitive taking days (>3 days) to complete. Additionally any changes in parameterisations would require the entire process to be setup again multiplying those days by the number of parameterisations required. Within EO4Vistrails the processing took four (4) hours and when combined with a number of parameterisations that are automated the entire process is reduced by an order of magnitude with respect to time. Although this would not stop the user from performing these tasks the ease with which they can be done within EO4Vistails is also a major motivating factor.

DST GEOSS

Description

The second system we describe is the use of EO4VisTrails in a distributed Sensor Web environment as part of the DST GEOSS project.

The project required extensive use of service-oriented standards-based interoperability to chain together heterogeneous, distributed data sources and processing functions. This project did not have the same big data and intensive computation requirements as the EO4HiTempo project described above. The functional requirements for the project included:

- Bindings to various OGC web service interfaces including the Sensor Observation Services (SOS), Web Feature Service (WFS) and Web Map Service.
- Binding to various file-based data sources including GeoTIFF, CSV, NetCDF and Shapefile.
- The use of the statistical capabilities in R.
- Binding to proprietary command line binaries, in this case a black box written in Cobol.
- Extensive use of map overlays and various feature operations including bounding box and intersection tests.

The above description of the project led to the following non-functional requirements:

- The need to integrate heterogeneous tools and languages.
- The ability to wrap arbitrary code, such as a black box process, and provide inputs and outputs.
- Extensive GIS functionality for handling proper representation of geospatial data, including projections etc.
- Standards-based approach to data access.
- Rapid deployment due to ease of integration.

Results and lessons learnt

The DST GEOSS project completed successfully with six (6) use cases being demonstrated that showed the utility of EO4VisTrails. The use of EO4VisTrails allowed for the rapid deployment of the use cases as a result of increased integration ability once the neccesary tooling had been completed. In addition the use of reusable components in the form of building blocks of a workflow linked to standards-based web interfaces and standards-based data interchange formats gave a lot of benefit, after the initial additional cost of development according to the standard had been covered. The benefits took the form of the use of open standards which allowed us to use a components off the shelf (COTS) with very little if any - modification. One other advantage of the approach taken was the easy integration of black-box processes, such as external, user-provided FORTRAN modules, into a wider system without needing access to the internals of the black box.

In summary, the scientific workflow approach to integrated GIS and standards-based approach within the DST GEOSS project resulted in a rapid deployment due to ease of integration as well as due to reuse. In addition there was an increased return on investment due to standards-based approach to scientific workflows. Also the scientific workflows approach resulted in increased access to COTS and the successful and easy wrapping and integration of existing black box tools.

Current EO4VisTrails projects

We also have a number of projects currently being worked on, which have EO4VisTrails as either an enabling or core component. These projects are dicussed below.

Firstly a project in which EO4Vistrails is being applied is the EO2Heaven is an FP-7 project that is focused on human health and environment proxies that affect that health. EO4VisTrails has already shown some utility in the EO2Heaven project where the automation of highly repetitive tasks allowed the project scientists to focus more on the science and far less on data handling and processing.

Secondly the CLUVA project, also an FP-7 project, with primary focus on climate change data will also use EO4Vistrails. Here standards such as OpenDAP and NetCDF CF-1 are prevalent. Many of the requirements for this project are already found in EO4VisTrails [7].

Another project where EO4Vistails is being used is SWEOS a CSIR project focused on the monitoring of fresh water quality in inland dams and lakes using remote sensed data. In addition the AMD project, also a CSIR project, will use EO4Vistrails and is focused on various aspects of acid mine drainage with a unique requirement for the integration of GIS and physical simulations. We look forward to the successful deployment of EO4VisTrails within the remainder of these and other projects.

Summary and conclusions

EO4VisTrails addresses the problem of “going beyond GIS” by integrating it with specialised toolsets such as statistical packages and algorithmic scripting languages. In addition, a standards-based approach provides much value and return on investment. In summary the Scientific Workflows approach has been shown, through successful application, to be a viable and useful approach to the challenge of geospatial and remote sensed data handling and processing.

The current GIS community has not caught up with other scientific fields in terms of the use of scientific workflows in day-to-day activities. However once the tooling in the form of the required components and functionalities is in place with the use of a standards-based approach, and once scientists and other GIS practitioners realise the massive benefits in terms of productivity and added value, scientific workflow tools will become increasingly used in the spatial data users workplace.

References

[1] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, S. P. Callahan, and H. T. Vo: “Vistrails: Visualization meets data management.”, *In ACM SIGMOD*, Chicago, Illinois, USA, 2006.

[2] A. Barker and J. Van Hemert: “Scientific workflow: A survey and research directions.”, *Parallel Processing and Applied Mathematics*, pp. 746-753, 2008.

[3] K. Chiu, M. Govindaraju, and R. Bramley: “Investigating the limits of SOAP performance for scientific computing.”, *In Proceedings of HPDC-11*, pp. 246-254, 2002.

[4] G. McFerren, T. L. van Zyl, M. van Der Merwe, and M. du Preez: “User Requirements for Sensor Web based Scientific Workflows in the Cholera Research Domain.”, *In 2008 IEEE International Geoscience and Remote Sensing Symposium*, Boston, USA, pp. 136-139, July 2008.

[5] Z. Meglicki: “Advanced Scientific Computing.”, 2001.

[6] J. J. Rehr, J. P. Gardner, M. Prange, L. Svec, and F. Vila: “Scientific Computing in the Cloud.”, *arXiv*, 2008.

[7] A. Vahed, F. Engelbrecht, I. Simonis, M. Naidoo, and T. L. van Zyl: “Harnessing Cyber-infrastructure for Local Scale Climate Change Research in Africa.”, *In IIMC (International Information Management Corporation)*, pp. 1-13, 2012.

[8] M. Vallisneri and S. Babak: “Python and XML for Agile Scientific Computing.”, *Computing in Science & Engineering*, 10(1), pp. 80-87, 2008.

[9] F. Van Den Bergh, K. J. Wessels, S. Mite, and T. L. van Zyl: “HiTempo: a platform for time-series analysis of remote-sensing satellite data in a high-performance computing environment.”, *Transactions of Remote Sensing*, 33(15), pp. 4720-4740, 2012.

[10] T. L. Van Zyl, G. McFerren, and A. Vahed: “Earth observation scientific workflows in a distributed computing environment.”, *In FOSS4G*, 2011.

[11] U. Yildiz, A. Guabtni, and A. H. H. Ngu: “Business versus Scientific Workflows: A Comparative Study.”, *2009 Congress on Services – I*, (March), pp. 340-343, July 2009.

Contact Terence van Zyl, CSIR, Tel 012 841-3460, tvzyl@csir.co.za