ADDING TEMPORAL DATA ENHANCEMENTS TO THE ADVANCED SPATIAL DATA INFRASTRUCTURE PLATFORM

Bolelang Sibolla[1], Terence Van Zyl[2], Graeme McFerren[2], Derek Hohls[2]

1. CSIR Meraka Institute, Bolelang Sibolla bsibolla@csir.co.za
2. CSIR Meraka Institute

KEYWORDS: Web mapping, SDI, Earth Observation Data infrastructure, Geonode, Temporal data

ABSTRACT

Users of Spatial Data Infrastructure (SDI) increasingly require provision to data holdings beyond the traditional static raster, map or vector based data sets within their organisations. The modern GIS practitioner and Spatial Data Scientist are required to work with data that not only has a spatial component but also a temporal component. The traditional SDI has been well suited for providing capabilities relating to governance, discovery, exploration, access and basic collation of static data sets. Across the world, Geonode is emerging as a framework for building spatial data infrastructures as it covers all of these aspects of traditional SDI. However, there is an increasing need for the capabilities of Geonode to be extended to data sets with a temporal component. Geonode is an Open Source Platform that was originally developed by the World Bank in collaboration with OpenGeo in response to the World Bank's need for an SDI on which to build a disaster management platform for reducing risk in Central America. Geonode has since been adopted in Europe and America as well as South Africa, with a growing user base. Similar to other open source programs, a number of enhancements have since been developed. Based on the realisation that more and more scientists are using and acquiring geospatial data, and the importance of time series in geospatial research, we have developed enhancements to the Geonode platform that enable the handling of temporal data. These tools enable the user to discover, explore, visualise and access temporal data. This paper begins with a discussion of the traditional and current SDI practices, in terms of governance, discovery, exploration, access, and collation of data. We review the Geonode platform and discuss its suitability and compliance with the requirements of a Spatial Data Infrastructure. Methods used for developing the enhancements to Geonode are discussed. The results of the project are presented and challenges and further work in this project are discussed.

INTRODUCTION

Increasingly users of Spatial Data Infrastructure (SDI) require provision to data holdings beyond the traditional static raster, vector or in-situ point based datasets available within their organisations. The modern GIS practitioner and Spatial Data Scientist are required to work with data that not only has a spatial component but also a temporal component.

Traditional SDI has been well suited to providing capabilities relating to governance, discovery, exploration, access and basic collation of static data sets. For instance the capabilities provided by the popular Geonode platform cover all of these aspects of traditional SDI. However, there is an increasing need for these capabilities to be extended to data sets with a temporal component.

This paper begins with a background discussion about SDI. Following that, Geonode as an open source based SDI platform is introduced and the short comings of Geonode as it stands are discussed setting a background of why the enhancements that we made were necessary. The development of the temporal data enhancements is then presented and results discussed. The paper closes off with a discussion of the result achieved, and recommendations on future work are made.

SDI BACKGROUND

The GSDI defines an SDI as the sum of technology, policies, standards, human resources and related activities that facilitate the acquisition, processing, distribution, use, maintenance and preservation of geospatial data, throughout government, private sector and academia (Nebert, 2000; Groot and McLaughlin, 2000; Maguire and Longley, 2005). Following Masser (1998) SDI's are developed to enable users to discover, explore and use geospatial data according to the needs of their applications. A number of discussions around the definition and development of SDIs can be found (Groot and McLaughlin, 2000; Williamson et al, 2003; Cooper et al, 2007), however most authors agree on aspects and components to be considered in the development of an effective SDI, and these can be found in the GSDI cookbook (Nebert, 2000). Amongst others a good SDI must facilitate data discovery, visual exploration of data, as well as data access and delivery (Nebert, 2000; Alders and Moellering, 2001; Crompvoets et al, 2004).

Data Sharing in an SDI Environment

It has been deemed impractical to expect centralisation of data holdings as the production of data has escalated in recent years. Data is housed and owned by different organisations; therefore the focus has shifted to distributed networks of data holdings that can communicate. Ability to share data requires harmonisation and interoperability of the data. Albrecht (1999) provides a definition and detailed discussion into the concepts of Interoperability with respect to an SDI. Albrecht's discussion provides justification for the need for standardisation to govern, facilitate and aid the data sharing processes in an SDI.

The internet is continuously growing as an effective means to share data across multiple networks and data holdings, enabling a wide range of geospatial data users to be reached. Standardised web services have therefore been used to facilitate this movement and transfer of data across multiple platforms. Crompvoets et al (2004) and Mohammadi et al (2010) provide further, detailed clarifications into interoperability issues with respect to web services.

GEONODE AS AN ADVANCED SDI PLATFORM

Why Geonode and Background

Geonode was developed by World Bank in corporation with OpenGeo as an underpinning SDI for the CAPRA program, a disaster risk management project for Central America. It is an open source based platform. The decision to make Geonode open source was in order to allow it to be improved and expanded at later stages. A study of other SDI's was undertaken by the World Bank team to investigate whether they could meet their requirements. The findings of the team were that most SDIs were lacking in terms of effective data sharing and collaboration (Pickle, 2010):

What is Geonode?

Geonode is based on open source components: Geoserver, PyCSW/GeoNetwork, Django and GeoExplorer/GeoExt. Together these provide a platform for spatial data discovery, access, exploration, summed up with a view on data access permissions for the security and integrity of data. It also has a social networking component in order to allow quick and easy well known communication amongst the users (Pickle, 2010). Geonode is highly interoperable and can be easily integrated into existing platforms, for example QGIS has a Geonode plugin.

Geonode has been used successfully across the world, and most notably in this context, it has been applied in Malawi, Kenya and Mozambique. Kenya, in collaboration with the World Resources Institute (WRI) developed a program called Virtual Kenya. The project was in order to provide access to the countries important datasets and maps, with a view to encourage and enable the knowledge users as well as ordinary users to develop interest in geospatial data. The World Bank's Global Facility for Disaster Reduction and Recovery organisation (GFDRR) developed two instances of Geonode in Malawi and Mozambique, MASDAP and Moz-ADAPT, respectively. The purpose of the Geonode projects was to empower the two countries to better understand and be prepared for activities that result from climate change. The emphasis was on natural disaster reduction and resilience in a changing world. The Malawian project provides weather and infrastructure data, whilst the Mozambican project also includes modelled climate change projection data both historical (from 1980) and projections up to 2100.

Principles of Geonode:

According to Pickle (2010) Geonode, and hence its components, is based on the following key principles:

- To promote collaboration amongst geospatial scientists as users, by making it easy to share data, social networking (for example adding comments, rating datasets) and allowing connectivity with multiple

Geonode instances.

- Distribution: automatic metadata creation on uploading of a data file and search via catalogues.

- Cartography: this can be referred to in general terms as the data visualisation component. It provides a mapping capability that allows for styling of data, saving maps and sharing.

- Data collection: providing a mechanism and an interface for uploading of data.

Main Components of Geonode Explained:

The three main components of Geonode, namely PyCSW, Geoserver and GeoExplorer are discussed below followed by a brief discussion about the development framework.

- **Geoserver**: An open source geospatial data server that allows users to edit and share geospatial data. It is able to publish data using the OGC WMS, WFS, WCS and WPS standards. It allows for editing of data using the SLD (Stylised Layer Descriptor) which is presented in XML (Extensible Markup Language). Geoserver can serve both raster and vector data as well as data from other databases such as PostGIS. Geonode uses Geoserver as it's main geospatial data server.

- **PyCSW:** is a python based implementation of the OGC's Catalogue Service for the Web (CSW) server. It allows for the discovery and publishing of metadata for geospatial datasets. It can be deployed as a standalone server or can also be embedded in other applications. Geonode uses PyCSW for harvesting and serving metadata, in order to make data searchable.

- **GeoExplorer**: is a JavaScript based web mapping application, which is built on GeoExt. The map application allows users to interactively navigate, organise and analyse geospatial data. It uses the OGC's WMS to collect and arrange geospatial data. GeoExplorer is used as the data viewer for geonode, for previewing data and also for map compositions.

- **Django Web Development Framework**: is a python based web development framework, which follows the Model-View-Controller architecture pattern. The main purpose is to ease web development and its strong point is that the components need to be reusable and pluggable. It also implements web security through its administrative interface. This allows for different users to manage use rights of components.

TIME SERIES EXPLORATION IN GEONODE

As discussed in earlier sections, many geospatial and earth observation scientists depend on spatio-temporal data to do their analysis in their domain specific projects. This data has until recently not been catered for in SDIs. The OGC provides a couple of standards for dealing with time series data. The focus of this paper will be on the OGC's Web Map Service Time (WMS-T) and adding support for this to Geonode.

WMS-Time is a time aware extension of the Web Map Service (WMS) that was discussed earlier. According to Kolodziej (2004), the main advantage of WMS is that software that uses this standard is able to pull map layers from multiple other conforming web applications, and present them in an aggregated WMS. This data may not necessarily be similar in map scale, map projection or coordinate system, and may also result from geoprocessing services offered by other WMS servers (De la Beaujardiere, 2006). Figure 1 shows the flow of information from database to web client facilitated by WMS service.

WMS supports the GET method of the HTTP internet protocol and has three main requests that can be sent from the client to the server, namely, GetCapabilities, Getmap, GetFeatureInfo. The GetCapabilities in summary, requests to get service or layer metadata, the client asks the server for the layers that are available to a service, what map projection system and data output formats are available; the server presents the response as an XML document. In the case of the Getmap request, the client asks for a map providing specifics such as projection, format, bounding box. These would have normally been acquired from the GetCapabilities request that a client issues before requesting for data. The GetFeatureInfo request: the client asks for attribute information about the features on the map layers. This request is only available to services that advertise themselves as queryable. (Becker, 2009; De la Beaujardiere, 2006)

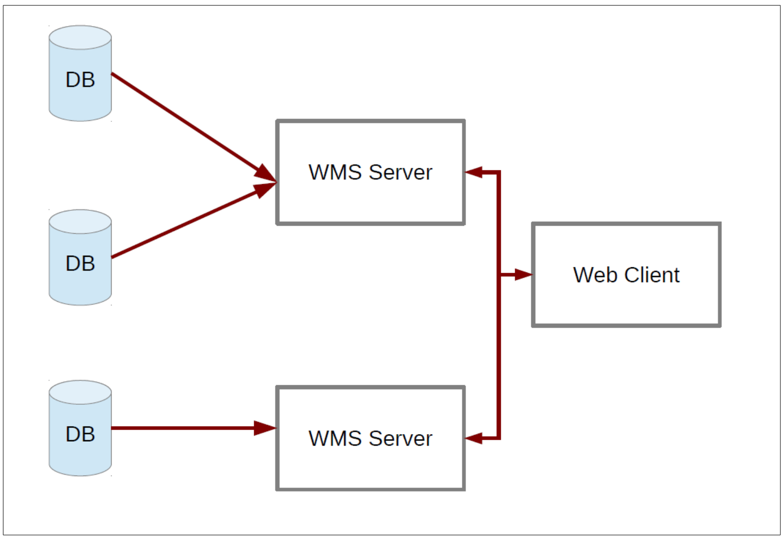

Figure 1. The flow of information from database to web client facilitated by WMS service.

If data is available for a phenomenon such as weather in the same geographical location for multiple time steps, such as hourly rainfall data, it can be represented in a WMS server as a WMS-Time. The time attribute will need to be configured in a way that the WMS will be aware of it, these time format specifications are provided in annex C and D of the specification document (De la Beaujardiere, 2006). For example:

```
name="time" units="ISO8601" default="2003-10-17">1996-01-01/2003-10-17/P1D
```

The time units are in ISO8601 and the format shown above indicates the start and end times of available data. If the time is properly configured, the WMS server will be able to provide available times in the GetCapabilities response document. When requesting a map using the GetMap request the client can include the time instance they require as part of the request URL.

Using the WMS Standard in Geonode

Data is uploaded into geonode in two ways, either directly through Geonode or through the linked instance of Geoserver. Once data has been uploaded using the two mechanisms discussed, a record of the data becomes available in the Geoserver instance of Geonode. The Geonode client is then able to request data from the Geoserver data store using the WMS mechanism described above, however it is not able to interpret the time based layer. When presented with a WMS request for a layer that is time aware Geonode returns data for the last available time stamp. This inability to display time based data resulted in the modifications presented below.

Currently uploading a time series dataset into Geonode is done through Geoserver. Geoserver is able to upload raster time series data using the ImageMosaic plugin. The plugin takes individual raster images that are georeferenced and cover the same area and named according to their timestamps e.g ***Pretoria_20091001.tif***. These images are then transformed into a single time queryable layer. It is worth mentioning that the ImageMosaic plugin can also be used to create a mosaic of spatially overlaping areas. When Geoserver registers a WMS-Time dataset through the Image Mosaic plugin, the metadata is sent to Geonode via models that also harvest the metadata information, and at the same time send it to the catalogue service PyCSW. It was realised at this stage that the WMS response from Geoserver already provides the desired WMS-time information, however this information was unused in Geonode.

Time Series Based Extensions to Geonode

Since the information about the WMS-Time is already passed into Geonode from Geoserver, the next step was to harvest this information and create a dynamic visualisation of the time stamps within a layer. An enhancement plugin in the form of an animation widget was added to the Geonode's GeoExplorer viewer to this effect. This widget was written such that it gets activated automatically in the viewer when the time variable is detected in the layer's metadata. The client does not see the time based animation widget in the viewer if there is no time series data available. Since GeoExplorer is built using the

GeoEXT javascript framework, and the viewing tools are based on the GXP components, the plugin is written as a GXP tool that plugs into the GeoExt built GeoExplorer viewer. The code is thus written in javascript programming language.

Accessing Time Series Data from Other Servers

Another problem arises when one wants to load data from other WMS-Time servers. Geoserver connects to other WMS servers through the cascading WMS operation. However this does not allow the loading of WMS-Time and only brings back the last time stamp of the dataset. This is a problem because Geonode only displays and loads data from a single co-registered Geoserver. An alternative route for accessing time series data into Geonode had to be sought.

A web based implementation of the WMS server for multidimensional data, ncWMS was used. ncWMS is an extension of the OGCs WMS service, that is meant for multidimensional data including data with a time component. Unlike in Geoserver where data is stored in a mosaic of tiff images, ncWMS takes data stored in netCDF format which is CF (Climate and Forecast Metadata Convention) compliant. More information on ncWMS is provided in Blower et al. (2013). Methods for converting data between the netCDF and Tiff formats are well known within the geospatial community. With ncWMS a direct GetCapabilities request was able to be made for time series data from within Geonode to the respective servers. This is facilitated by the WMS standard being able to read the metadata that is in the known Climate and Forecast (CF) convention. The servers return a list of available time aware layers which can then be previewed on the customised viewer using the animation widget.

The WMS from the time aware layers can also be overlaid with data from the Geonodes geoserver to create an aggregated map that gives more information. The map can then be saved, published and shared with other users using Geonode's map publishing service.

The methods described above show the short comings of Geonode in terms of accessing and visualising time series data. The ways in which these shortcomings are overcome include the incorporation of functionality to read from ncWMS servers, as well as the use of Geoserver's Image Mosaic plugin for overlaying tiff images as bands in a single layer with time component. The viewer is then enhanced to support the visualisation of time series data.

RESULTS

The visualisation was tested using the European Centre for Medium Range Weather Forecasting's (ECMWF), relative humidity data. The data was acquired for a whole month cycle, for the Southern African region. The ECMWF data was acquired from their download server in netCDF, CF-1.5 compliant format. Which made it easy to put it behind our local ncWMS server as is. The server was then queried using the specific dataset URL that is provided by the server through a GetCapabilities request. The process is simplified such that the user only supplies the URL of the ncWMS server they want to query. Geonode is configured in this case such that a GetCapabilities request is performed in the background and a list of available data sets is then displayed. The user then clicks on the link to the dataset they are interested in and this is then displayed in the animation aware viewer for exploration.

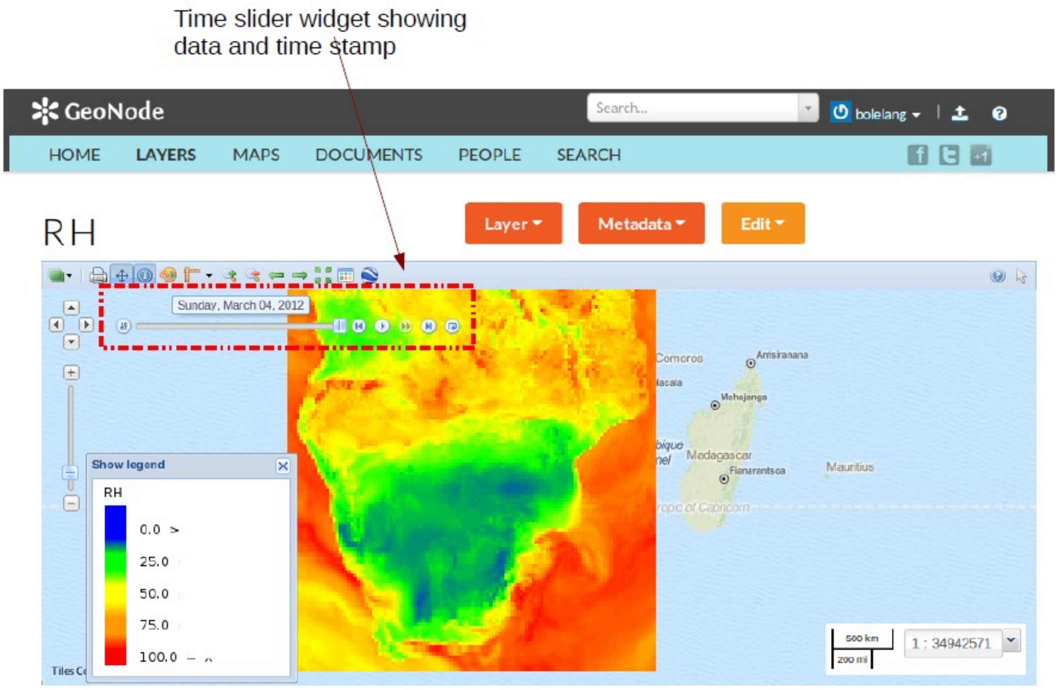

Figure 2. ECMWF Time series data displayed in Geonode

The same dataset was used to test the upload of data into Geoserver as a time aware ImageMosaic. In order to display this data in Geonode through Geoserver, it was first converted from netCDF to tiff files using an automated script. The tiff files were then loaded into Geoserver as an Image Mosaic. Geonode is then made aware of this data using an update layers command that notifies Geonode of new data that has been loaded into Geoserver. The linked PyCSW catalogue service loads all the necessary metadata from Geoserver for Geonode to be aware of. Once Geonode has been made aware of the time aware data, when the user views it in the Geonode GeoExplorer viewer, the animated time slider is activated and the user is hence able to explore all available time steps of the data. This is illustrated in figure 2.

CONCLUSIONS AND RECOMMENDATIONS

Further Work

The results of the time-series enabling process in Geonode so far have been presented in the section above. However in a complete SDI platform just viewing of data is not enough, the user may want to download the full time series data or parts thereof. The download function is under development currently and results of this will be presented in due course.

Conclusions

It was found that Geonode performed all the functions that are necessary for a successful SDI platform. However it did not cater for spatio-temporal datasets. This was the basis on which the enhancements discussed above were implemented.

Since Geonode is based on the Django Web development framework, the source code is well structured and hence easy to read. This ensured that it was easy to implement the new code changes, and test the additions that were made.

Once the additions have been fully tested the changes can be contributed back into the main Geonode source code for the community to adopt following a thorough review process.

REFERENCES

Albrecht, J. *1999*. Geospatial information standards. A comparative study of approaches in the standardisation of geospatial information. *Computers & Geosciences,* 25 (1999), pp. 9-24.

Alders, HJGL., Moellering, H. 2001. Spatial Data Infrastructure. *Proceedings of the 20th* International *Cartographic Conference*, Beijing, China, Vol. 4, pp. 2234-2244.

Blower, J.D., Gemmel, A.L., Griffiths, G.H., Haines, K., Santokhee, A., Yang, X. 2013. A Web Map Service implementation for the visualization of multidimensional gridded environmental data. *Environmental Modelling and Software*, (2013)47, pp. 218-224.

Berker, T. 2009. Visualizing Time Series Data Using Web Map Service Time Dimension and SVG Interactive Animation. *MSc Degree thesis*. International Institute for Geo-information Science. Netherlands.

Cooper, A., Moellering, H., Delgado, T., Düren, U., Hjelmager, J., Huet, M., Rapant, P., Rajabifard, A., Laurent, D., Iwaniak, A., Abad, P., Martynenko, A. 2007. An initial model of the computation viewpoint for a spatial data infrastructure. *23rd International Cartographic Conference*, Moscow, Russia, pp 10

Cooper, A.K., Moellering, H., Hjelmager, J., Rapant, P., Delgado, T., Laurent, D., Coetzee, S., Danko, D.M., Düren, U., Iwaniak, A., Brodeur, J., Abad, P., Huet, M., Rajabifard, A. 2013. A spatial data infrastructure model from the computational viewpoint*.* International Journal of Geographical Information Science*, (*27)6, pp. 1133-1151.

Crompvoets, J., Rajabifard, A., Bregt, A., Williamson, I. 2004. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographic Information Systems,* (18)7. pp. 665-689.

De la Beaujardiere, J. (eds). 2006. OpenGIS® Web Map Server Implementation Specification. *Open* Geospatial *Consortium Inc.*

Di, L., 2004. Distributed Geospatial Information Services-Architectures, Standards, and Research Issues. *The International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, Vol. XXXV, Part2, Commission II, 7p

Fitzke, J., Greve, K., Muller, M., Poth, A. 2004. Building SDIs with Free Software – the degree project. *9th International Symposium on Planning & IT*. Vienna, Austria

Groot, R., McLaughlin, J. (eds). 2000. Geospatial Data Infrastructure: Concepts, Cases and Good Practice, Oxford University Press, Oxford, UK.

Kraak, M.J. 2002. Current trends in visualisation of geospatial data with special reference to cartography. *Proceedings of the 22nd INCA Congress, Indian National Cartographic Association: Convergence of Imagery, Information and Maps*, Ahmedabad. *Indian Cartographer*, (2002)22, pp. 319 – 324.

Kohler, P., Watcher, J. 2006. Towards an Open Information Infrastructure for Disaster Research and Management: Data Management and Information Systems Inside DFNK. *Natural Hazards* (2006) 38, pp. 141–157.

Longhorn, R. 2005. Geospatial Standards, Interoperability, Metadata Semantics and Spatial Data Infrastructure. NIEeS *Workshop on Activating Metadata*, Cambridge, UK.

Luchtman, S., Hosein, P. 2014. Design And Specifications Of A Repository For Real-Time Open Data. *Proceedings of the 2014 ITU Kaleidoscope Academic Conference: Living in a converged world*- impossible *without standards.* pp. 105 – 110.

Maguire, DJ., Longley, PA. 2005. The emergence of geoportals and their role in spatial data infrastructure. *Computers, Environment and Urban Systems,* (29)1, pp. 3-14.

Masser, I. 1998. The first generation of national geographic information strategies. *Theme Papers GSDI* Canberra *Australia*. http://www.gsdidocs.org/docs1998/canberra/masser.html (accessed 14 July 2014)

Mohammadi, H., Rajabifard, A., Williamson, I. 2010. Development of an interoperable tool to facilitate spatial data integration in the context of SDI. *International Journal of Geographical* Information *Science*, (24)4, pp. 487–505.

Nebert, D.D. (ed). 2000. Developing Spatial data Infrastructures: The SDI Cookbook. http://www.gsdi.org/pubs/cookbook/chapter01.html#spatial

Kolodziej, k. (ed). 2004. OGC OpenGIS Web Map Server Cookbook. Open Geospatial Consortium, 1.0.2 edition.