

Masters Program in **Geospatial Technologies**



Sensor Data Visualization in Virtual Globes

Luis Enrique Rodríguez Pupo

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

Sensor Data Visualization in Virtual Globes

Dissertation supervised by

PhD. Alain Tamayo Fong

Dissertation co-supervised by

PhD. Pedro da Costa Brito Cabral (ISEGI)

PhD. Joaquín Huerta Guijarro (UJI)

March 2011

ACKNOWLEDGMENTS

I would like to recognize the help offered by my supervisors and especially to Alain Tamayo for his constructive advices about this work. I would also like to thank Arturo Beltran for sharing the initial code of the tool I have developed further. I am also especially thankful for the support of my classmates and professors through the whole master program. This topic and the master program have been specially challenging for me and the help offered by others have been fundamental for achieving my goals.

Sensor Data Visualization in Virtual Globes

ABSTRACT

With the recent developments related with sensors in matters of standardization and accessibility, valuable data covering different geographical subjects have become widely available. The applications that can leverage sensor data are still under development and there is much to do in this subject in the scientific community. Data visualization tools are one of the immediately relevant needs related with sensor data. Such tools would help to increase the understanding and exploration of the data from which many other fields can get benefits.

Virtual Globes are becoming increasingly popular in the society. The existence of several implementations and millions of users (scientific and no scientific) around the world are a proof of their increasing usability as a tool for representing and sharing geographical content.

In this document we present a generic tool for visualizing sensor data retrieved from SOS servers over the NASA World Wind virtual globe. For this, we started by creating a classification of sensor data that helps in defining possible visualizations for the different types of sensor data. Using this classification as a basis, we have implemented a set of visualization types to ease sensor data exploration. We also included analysis capabilities by integrating the SEXTANTE library in the visualization tool. The results of the analysis can be included in the virtual globe as part of the visualizations.

KEYWORDS

Sensor Web

Sensor Data Visualization

Virtual Globes

Data Analysis

ACRONYMS

3D – Three Dimensional

API – Application Programming Interface

Eclipse RCP – Eclipse Rich Client Platform

HTML – Hypertext Markup Language

O&M – Observations and Measurements

OGC – Open Geospatial Consortium

SDK – Software Development Kit

SOS – Sensor Observation Service

SWE – Sensor Web Enablement

SensorML – Sensor Modeling Language

TML – Transducer Markup Language

WMS – Web Map Service

WWJ – NASA World Wind Java

INDEX OF THE TEXT

ABSTRACT	iv
KEYWORDS	v
ACRONYMS	vi
INDEX OF THE TEXT	vii
INDEX OF TABLES	viii
INDEX OF FIGURES	ix
1. INTRODUCTION	1
1.1 Objectives	3
2. BACKGROUND	4
2.1 Virtual Globes.	4
2.1.1 NASA World Wind for Java	5
2.2 SOS, SensorML and O&M.	7
2.2.1 Sensor Observation Service (SOS)	8
2.2.2 GetObservation parameters	10
2.2.3 Filters especification	12
2.2.4 SensorML	13
2.2.5 Observations and Measurements (O&M).	14
3 SENSOR DATA CLASSIFICATION FOR VISUALIZATION	19
3.1 Data belonging to category I	20
3.1.1 One sensor	20
3.1.2 Multiple sensors.....	21
3.2 Data belonging to category II	22
3.2.1 Data that varies its spatial position	22
3.2.2 Data that varies with time	23
3.2.3 Data that varies in space and time	23
3.3 Other types	23

3.4	Summary.....	25
4	PROTOTYPE DESCRIPTION	26
4.1	General Description	26
4.1.1	Communication Library.....	29
4.1.2	GetObservation operation support.....	29
4.1.3	Data Handling and presentation	30
4.1.4	Integration with SEXTANTE library.....	32
4.2	Visualizations	35
4.2.1	Chart visualizations	37
4.2.2	Animation of observations.	39
4.2.3	Visualization of SEXTANTE results.	40
4.2.4	Visualization Control.....	43
4.2.5	Non-data related visualizations.....	44
4.3	Summary.....	45
5	CONCLUSIONS AND FUTURE WORK	46
5.1	Limitations.....	46
5.2	Future Developments.....	47
	REFERENCES.....	49

INDEX OF TABLES

Table 1:	GetObservation operation parameters	11
Table 2:	Specialized Observations for constant result observations.	15
Table 3:	Specialized Observations for variable result observations.	16
Table 4:	SEXTANTE input parameter types.....	34

INDEX OF FIGURES

Figure 1: Example of Scatter Plot Chart.....	22
Figure 2: Example of Difference Chart.....	22
Figure 3: Example of XY Chart.....	22
Figure 4: Time series chart.....	22
Figure 5: Examples of Analytic Surfaces.....	23
Figure 6 : Example of Contour Lines.....	23
Figure 7: Main components of the prototype.....	27
Figure 8: General composition of the prototype.	28
Figure 9: GetObservation wizard pages.	30
Figure 10: Structural View of the datasets.	31
Figure 11: GetObservation results view.	31
Figure 12: Dataset Handling View.....	32
Figure 13: SEXTANTE library user interface.....	33
Figure 14: Visualization wizard.....	36
Figure 15: Chart Preferences page.....	37
Figure 16: Time series charts with multi-axis support.	38
Figure 17: Scatter Plot for Temperature and Humidity.	39
Figure 18: Animation of one observed property.	40
Figure 19: SEXTANTE output selection.	41
Figure 20: Histogram produced by SEXTANTE.	42
Figure 21: Basic statistics created with SEXTANTE.	42
Figure 22: Buffers created with SEXTANTE.....	43
Figure 23: Visualization Control View.....	44
Figure 24: Visualization of the offerings advertized by a SOS server.....	44

1. INTRODUCTION

A sensor is a device that reacts to a certain stimulus and gives as output an electrical signal representing the magnitude of the stimulus. A further definition, geo-sensor, is a sensor in which the output can be geographically referenced (Craglia, et al., 2008) . The output of the sensors can have a varied nature, ranging from multi-band images to measurements of temperature and salinity levels.

The spectrum of existing geo-sensors is very wide and they have been in existence for many years. Developments of vast networks of geo-sensors have been put into practice in many projects; examples include the networks implemented in different projects by the World Meteorological Organization (WMO¹). To have a rough idea, according to the information available in the WMO website, it counts with more than 10000 manned and automatic surface weather stations, 1000 upper-air stations, over 7000 ships, more than 100 moored and 1000 drifting buoys, hundreds of weather radars and over 3000 specially equipped commercial aircraft measuring key parameters of the atmosphere, land and ocean surface every day (WMO, 2012).

Recent advances related to the geo-sensors under the OGC Sensor Web Enablement (SWE) initiative included various directions of development. The initiative focuses in important aspects such as interoperability, integration, data encoding and publication. These efforts led to the creation of a sensor web where the geo-sensors can be tasked, discovered and accessed through web standards (Craglia, et al., 2008). Another direction of work has been the specification of the characteristics of the sensor systems for enforcing high accessibility to sensor metadata and the self description of the sensor systems. Under this initiative, a set of standards have been defined for different subjects, most of them relevant to our work.

The presence of these standards has proven its effectiveness as they have encouraged initiatives for publishing and sharing sensor data. Nowadays there are many services relying on the SWE standards offering sensor data in internet, in many cases with big volumes of valuable data. Nevertheless, the value of the data is only subjective

¹ <http://www.wmo.int>

without the existence of tools able to help in exploring and reasoning about it. In Chapter 2 we address the relevant technologies to this thesis in a finer detail.

Regarding the ways of representing geographical data, the technological advances in different computing areas made feasible the representation of the Earth by giving a global view relying on existing imagery. The advent of *Virtual Globes* opened a new world of possibilities in terms of representation given their ability of providing with updated and accurate imagery of the world (Tuttle & D., 2008). As they provide geographical information in a way that is possible to perceive the 3D aspects of the geographical features, the virtual globes give a more ‘realistic’ perception of the world.

Most of the Virtual Globes support the representation of different types of content: images, videos, points, 2D/3D shapes, and dynamic content. These are fundamental tools for the representation of location-aware data. Data visualization, in general, always has the purpose of communicating the meaning of the data, a certain feature of it, or exposing relations existing between different datasets.

In fields as spatial problem-solving and decision-making the visualizations play a fundamental role for helping experts to solve problems. Therefore, the visualization methods have to encourage and ease the linking of the experts’ background knowledge with the facts exposed by the data. This responsibility has to be supported necessarily by the existing geographical analysis tools. The virtual globes by themselves are not an analytical environment, but a media for supporting visualization. In consequence, as pointed in (Andrienko, et al., 2007), the integration of these environments with existing visualization techniques and analysis tools is needed.

Characteristics of the sensor data available through the sensor web such as the high accessibility, interoperability, volume, heterogeneity and its spatial distribution make it ideal for data analysis.

1.1 Objectives

The objectives of this thesis include the following:

- Characterization and classification of sensor data with the aim of identifying appropriate visualization methods for each data category.
- Implementation of a generic prototype to visualize SOS-published sensor data on a Virtual Globe.
- Integration of the SEXTANTE library² to add spatial data analysis capabilities to the prototype for enabling the visualization of the analysis results.

The objectives have the ultimate goal of implementing a general tool for visualizing sensor data on the NASA World Wind for Java virtual globe, allowing the inclusion of analysis results as part of the visualization.

The rest of the document is structured as follows: the Chapter 2 includes a background containing the SWE technologies relevant to this work, a summary of the evolution and characteristics of Virtual Globes and the related work showing similar efforts. The Chapter 3 presents a classification of the sensor data and suggests a set of possible visualization types for each class. In Chapter 4 is presented a prototype application that relies in the previous classification as a basis for supporting a set of visualization methods. Finally, in Chapter 5 we include the conclusions of this work, its limitations and the future improvements that can be done to it.

² <http://www.sextantegis.com/>

2. BACKGROUND

In this chapter we introduce the main technologies related to our work. It includes mainly virtual globes, particularly NASA World Wind, and the technologies related to the SWE initiative. It is also included a section containing other works that have been done in the matter of sensor data visualization.

2.1 Virtual Globes.

According to Wikipedia a Virtual Globe is a “3D software model or representation of the Earth or another world” (Wikipedia, 2012). Virtual Globes provide the user with the ability of exploring the globe by changing the view, the angle position, and the altitude with respect to the surface. Usually, the virtual globes integrate imagery from different sources (Aurambout & Pettit, 2008). Depending on the scale the user is looking at the globe, imagery from different data sources can be presented.

Along the past two decades many virtual globes have been developed. In 1997 Microsoft released an offline virtual globe named *Encarta Virtual Globe 98* as a part of the Encarta 98 encyclopedia. Later in 1999 Cosmi Corporation released the *3D World Atlas*, a virtual globe that included many new features and integrated information of the countries using data from the CIA World Fact Book 1999 (CIA, 1999). By the middle of 2004 the NASA World Wind³ virtual globe was released and was the first announced online virtual globe. Later, in 2004 Google Earth⁴ was released, a successor of the Earth Viewer 3D software, after the acquisition of the company Keyhole Inc. by Google. After the public release of Google Earth, the interest of the public by virtual globes, geospatial technologies and its applications has experimented a huge increase.

The virtual globes exhibit different characteristics that include aspects as the resolution and source of the imagery, licensing policy, underline technology, ability of including different types of content, extensibility, etc. As source for imagery,

³ <http://worldwind.arc.nasa.gov>

⁴ <http://www.google.com/earth/index.html>

virtual globes use satellite and aircraft images, obtained from both public and private providers. The imagery can coexist with different temporal and spatial resolution in a given view of the globes. Common sources of imagery include Landsat (Landsat Program, 2011), with resolutions of 30 meters⁵, Spot Image (Wikipedia, 2011) with resolutions ranging from 2.5 meters to 10 meters per pixel, and USGS imagery⁶.

For aircraft imagery usually are used local companies that provide high resolution, low-altitude aerial images. These are later integrated in the databases feeding the virtual globes. These aircraft images have a more local coverage, and therefore it is possible to observe higher resolutions in big cities and other points of interest in the globe. Besides these elements, other data used consists in digital elevations models and bathymetry images. The later is used in case the virtual globe incorporates images of the seafloor, a feature that is not commonly supported.

Many of these aspects are relevant for choosing a virtual Globe for a giving representation. Depending on the intended use of the globe, some features can be determining. The general use of the globes by regular users does not require the use of many features, nevertheless for scientific purposes, the requirements could be totally different. In the scientific community, features such as the extensibility, KML⁷ representation capability, and temporal and spatial resolution of the images acquire a higher importance.

2.1.1 NASA World Wind for Java

NASA World Wind is an open source virtual globe developed by NASA and released in 2004. Initially NASA World Wind was implemented to be used on the Microsoft Windows⁸ platform. The initial implementation used .NET⁹ and DirectX¹⁰ technologies. Later, in 2007 with the version 1.4 it was released a multiplatform version, implemented in Java, named NASA World Wind for Java (WWJ).

⁵ Refers to the visible spectrum.

⁶ <http://www.usgs.gov/pubprod/aerial.html>

⁷ <http://www.opengeospatial.org/standards/kml>

⁸ <http://windows.microsoft.com>

⁹ http://en.wikipedia.org/wiki/.NET_Framework

¹⁰ <http://es.wikipedia.org/wiki/DirectX>

As data sources, WWJ uses imagery from Landsat¹¹, Blue Marble(NASA, Earth Observatory, 2012) imagery, USGS aerial imagery and data from community contributors (using Zoomit add-on) (World Wind Central, 2009). For terrain elevation data, WWJ uses SRTM (SRTM30Plus/SRTMv2/USGS NED)(NASA, Jet Propulsion Laboratory, 2009) terrain data that also includes bathymetry data.

Apart from these datasets, WWJ allows the inclusion of many others available through the use of plug-ins that can be incorporated to the SDK. Examples of such types of data are shape files, WMS data, GLOBE data (The Globe Program, 2012) and KML documents.

WWJ was designed in a way that it is mostly an SDK, not an application per se. It means that WWJ can be embedded and reused in other applications¹², including Java Applets¹³. Due to the fact that WWJ is open source, there exist many different contributions and branches to its code. Some of them are Dapple (Dapple, 2012), and SERVIR-VIZ (IAGT, 2012). NASA World Wind for Java contains a wide range of features for displaying, interacting and extending it.

A main feature of WWJ is the richness of the API offered, it allows the customization and interaction with almost every aspect of the virtual globe. At a high level WWJ allows the creation of geometries such as polygons, lines, points, and textual and HTML 5¹⁴ content. It is also possible to include many other forms of content such as videos and audio files. Other possible representations include the utilization of 3D elements, such as extruded polygons, and different kind of volumetric shapes (i.e. cones, cylinders, cubes, etc). Besides of this, WWJ allows the access to a lower level functionality for performing a highly customized rendering on the surface using the OpenGL library (The Kronos Group, 2012). The SDK contains more than one hundred examples exploring most of the functionality offered, which are easily extensible and adaptable to the needs of the user. Relying on this API, there is a wide variety of plug-ins that allows incorporating new functionality produced by the WWJ community.

¹¹ <http://landsat.gsfc.nasa.gov/>

¹² Applications using Java technologies.

¹³ See <http://java.sun.com/applets/>

¹⁴ <http://www.w3.org/TR/html5/>

All these characteristics make this virtual globe ideal for visualization of location-aware data. The extensibility and the possibility of embedding the virtual globe in a custom application allow the reuse of its functionality for context-specific purposes. Having full control over the content visualized and fully interaction capabilities is a very important feature, as the visualizations would not need to be restricted to a predefined set of visual representations. Besides, the user experience regarding the interaction with the visualized elements can be improved and adapted to the needs of particular visualization methods.

2.2 SOS, SensorML and O&M.

As mentioned in the introduction, the geo-sensor technology has experienced big advances in term of standardization, accessibility and interoperability. This is mostly due to the efforts in several projects carried out by the Open Geospatial Consortium (OGC) organization.

The OGC under the SWE initiative has developed a set of standards and protocols (Percivall, Reed, & Davidson, 2007) for accessing, publishing, and describing sensors and sensor data. The standards address different aspects of the interaction with sensor systems and sensor data. The main standards developed comprise the following (Botts, Percivall, Reed, & Davidson, 2007):

- **Observations & Measurements (O&M):** This standard comprises the data models and XML encodings for sensor data, grouped as observations and measurements. The standard includes the encodings of different aspects of the observations, the values of the observed property, and the time when the measurement was performed (Cox, 2010).
- **Sensor Model Language (SensorML):** Contains the standard model and XML schemas for describing processes and the sensor systems. The processes refer to the process taking place during the acquisition or creation of the data (Boots, 2007).
- **Sensor Observation Service (SOS):** Describes the protocol and the web services interface for accessing and publishing sensor data. This standard

regulates the way of publishing the data in favor of the interoperability of the services (Na & Priest, 2007).

- **Sensor Planning Service (SPS):** Web services interface to evaluate the feasibility of collecting data from sensor systems and submit collections requests (Simonis & Echterhoff, 2011).
- **SWE Common Data Model:** Defines low-level data models for exchanging sensor related data between nodes of the SWE framework (Robin, 2011).
- **SWE Service Model :** Defines data types for common use across SWE services (Echterhoff, 2011).

The Observations and Measurements standard is probably the most important for this work as it contains the specification related to how data provided by a certain sensor system is going to be encoded for ensuring the interoperability. It encodes most of the information related to data, its quality, dimensions, temporal components, units, etc.

2.2.1 Sensor Observation Service (SOS)

The SOS specification defines a protocol and a set of schemas for submitting requests to SOS services. A SOS service is the entry point to access sensor data. The specification is comprised of different profiles that, when implemented by the SOS server, enables the exposure of different functionality. Through the use of profiles, the functionality exhibited by a SOS server is split in a way that the operations of accessing and publishing the sensor data remain loosely decoupled. This act as a mechanism for allowing SOS server implementations to be merely providers of sensor data, or, in other case allow the publication of sensor data by registered sensor systems. The three existing profiles are: *core*, *transactional*, and *enhanced*.

The core profile is the only mandatory by the SOS specification and crucial to our purposes of visualization, given that it provides the access to the sensor data published in a SOS server. It contains the following operations:

- ***GetCapabilities:*** This operation is the entry point to the service and provides information about the available observations, as well as the filter capabilities provided by the server. Those capabilities are relevant for the process of

visualization, if we consider the data selection and manipulation as part of it. The result of executing a *GetCapabilities* operation is a set of observations, grouped by a certain criteria, the so called *Offerings*. The grouping criteria, usually takes into account the spatial distribution of the sensor systems involved, which is in many senses convenient. The consequence is that the offerings group the sensors in constellations, a term used to remark that there is a criterion of similarity between the observations belonging to one offering. This is a useful for visualization of the spatial distribution of the sensors in form of groups so that the user can directly address one of these groups for further sensor data retrieval, visualization or analysis.

- ***DescribeSensor:*** Allows obtaining detailed metadata information about a specific sensor, the response to this type of request is encoded in either SensorML (Boots, 2007) or TML (Transducer Markup Language) (Havens, 2007). This data could be useful for providing information about the sensor during visualization.
- ***GetObservation:*** This operation is used for accessing the observations kept in the SOS server. Depending on the capabilities of the server, this operation can support different kind of filters. Filters help in the data selection process by eliminating data irrelevant for the user. Three types of filters are usually supported. Temporal filters are used for selecting observations performed in a time period or a particular instant. Spatial filters allow the selection of observations depending on their spatial properties. The last type of filters, comparison filters, allows filtering the observations according to the values of a given property.

The transactional profile contains operations for registering a sensor system and inserting observations in a SOS service. The support of these operations by the SOS server is advertised in the *GetCapabilities* response. This profile has a lower relevance regarding data visualization as it is not involved in any process related with the data access. The operations included in this profile are:

- ***RegisterSensor:*** Allow the registration of a sensor or a sensor system in the SOS server in order to perform insertions of observations on it. In this

process it is necessary to provide a sensor description, either in SensorML or TML that could be retrieved in the future by a *DescribeSensor* operation. It is also necessary to include a template of O&M document that is used during the insertion of observations.

- ***InsertObservation***: Performs the insertion of an observation in the SOS server once the sensor has been registered. The request must include an O&M document with the observation values.

The last profile is the enhanced profile. It comprises operations that improve the core profile by allowing punctual selection of observations by its Id or retrieving partial results based on a previously performed *GetObservation* operation.

- ***GetObservationById***: performs a *GetObservation* based on an observation Id.
- ***GetResult***: This operation is intended to help reducing the overload incurred while sending repeatedly *GetObservation* operations to the SOS server. It works in collaboration with *GetObservation* operation, which includes a mode for establishing a template in the SOS server. The subsequent requests performed by using *GetResult* only have to submit a subset of parameters to the server. In the server the parameters submitted are then used to fill the template, assembling a normal *GetObservation* request parameter. In this way, the requests can be lighter and reduce the bandwidth usage while asking for observations.
- ***GetFeatureOfInterest***: Allows obtaining the feature associated to an Id specified in the operation parameters. This operation also allows filtering either by time or by spatial properties of the feature.

2.2.2 GetObservation parameters

For executing a *GetObservation* operation in a SOS server it is necessary to configure a set of parameters that mainly depend on the response to *GetCapabilities* operation. The most relevant parameters are listed in the Table 1. The table shows that most of the values admitted by these parameters are previously advertized.

The *GetObservation* operation has a set of mandatory parameters that include the offering and observed properties. Some of these parameters are placeholders for including different types of filters in the request. The parameter *eventTime* admits elements of type *ogc:TemporalOps*, the base class of the temporal filters. The parameters *featureOfInterest* and *result* are used to include spatial and property-based filters.

Table 1: GetObservation operation parameters

Parameter	Description	Cardinality/Type
offering	Specifies the offering URI advertised in the GetCapabilities document. All of the following parameters are dependent on the selected offering.	One (mandatory)
eventTime	Specifies the time period(s) for which observations are requested. This allows a client to request observations from a specific instant, multiple instances or periods of time in the past, present and future. The supported range is listed in the selected offering capabilities.	Zero or many (optional)
procedure	The procedure parameter specifies the URI of sensor system(s) for which observations are requested.	Zero or many (optional)
observedProperty	Specifies the URI of the phenomenon for which observations are requested.	One or many (mandatory)
featureOfInterest	Specifies the feature for which observations are requested. This can either be represented by a reference to a feature ID advertised in the capabilities document or can be a spatial constraint.	Zero or one (optional)
result	Provides a place to put in OGC filter expressions based on property values. This instructs the SOS to only return observations where the result matches this expression.	Zero or one (optional)
responseFormat	Specifies the desired resultFormat MIME content type for transport of the results (e.g. TML, O&M native format, or MPEG stream out-of-band). The supported output formats are listed in the selected offering capabilities.	One (mandatory)
resultModel	Specifies the name of the root element of an O&M Observation or element in the appropriate substitution group. The resultModels supported by a SOS server are listed in the resultModel element of the offerings in the service capabilities.	Zero or one (optional)
responseMode	Specifies whether results are requested in-line, outof- band, as an attachment, or if this is a request for an observation template that will be used for subsequent calls to	One (mandatory)

	<p>GetResult. This is provided to allow the client to request the form of the response. The value of resultTemplate is used to retrieve an observation template that will later be used in calls to GetResult. The other options allow results to appear inline in a resultTag (inline), external to the observation element (out-of-band) or as a MIME attachment (attached).</p>	
--	--	--

Other parameters are used for controlling how the results are going to be encoded and delivered to the client application executing the *GetObservation* operation.

2.2.3 Filters especification

The SOS servers can contain big volumes of data, and therefore without having filtering capabilities, it might not be feasible to retrieve the data from SOS servers. Therefore it is essential the use of filters in the *GetObservation* operation, though in many cases not all types of filter are supported.

The OGC created a standard for defining the encoding of the filters (Vretanos, 2010) that is used not only in the sensor related standards but in many other specifications. Filters fall in three main categories having specific operators in each of them. The existing categories are: the temporal filters, spatial filters, and the property-based filters.

The temporal filters allow specifying if an observation should be retrieved depending on a certain temporal relation with a parameter specified in the filter. There are temporal filters that express the relation of the observation sampling time with respect to an instant or a temporal period. Such relation is defined by a set of operators that include: inclusion of the observations retrieved in a certain temporal period, exact matching, precedence relations of the time with an instant, etc. The variety of operators included in the specification is wide enough to meet all the needs regarding temporal filtering.

The spatial filters, in a similar way, allow defining the spatial relationship of a certain property of the requested observations with the parameters of the filter. The existing operators comprise spatial relations such as inclusion in a bounding box,

overlapping, touching, crossing, containing, etc. These filters allow focusing on a specific spatial context for retrieving the observations from a given offering.

The last type of supported filters is the property-based filters. These filters are even more flexible in the sense that they allow filtering by the value of a certain property of the observations, specified in the filter parameters. The operators included in this category enable to compare the values of the property specified with a given value. Possible relations, for just mentioning some are: equality, precedence (i.e. if the property value is greater than the value specified in the filter) and range. An even more flexible variant of these filters allow the validation of the value of the property against a regular expression.

2.2.4 SensorML

As mentioned in section 2.2.1, the result of a *DescribeSensor* operation is encoded in SensorML or TML. SensorML is a standard to describe sensor systems. It also includes the definition of common data types present in other specifications of the SWE. SensorML specification does not contain any dependency with other schemas included in the SWE specifications, and therefore is intended to be used in other contexts. Within the SensorML specification, the sensors are treated as processes, which is an abstraction made in favor of the genericity of the specification.

Among the purposes of the specification are the support of explicit description of the sensor systems and processes, allowing the chaining of processes and the geo-location of the resulting observations (measured data). It is also intended to provide information about the performance of the sensor systems (such as accuracy or threshold).

The processes in SensorML are characterized as entities receiving one or more inputs, a set of well defined methods, and one or more outputs resulting from the application of the methods to the inputs. This conceptualization allows expressing transformations in a generic way, ranging from acquisition processes (made by transducers, or any other sensors) to coordinates transformations.

SensorML is the key of autonomous and intelligent sensor webs as it describes the functionality of the sensors in a way that they can be orchestrated by other services to obtain specific results.

2.2.5 Observations and Measurements (O&M).

Observations and Measurements is a standard for defining a general model for representing observations. The XML implementation (Cox, 2011) of this standard defines the encodings for observations in order to standardize their representation in XML.

Fundamental concepts for understanding O&M and sensor-related specifications include (Cox, 2010):

- **Feature of Interest:** A feature of interest is an abstraction of real-world phenomena having a set of properties that can be observed.
- **Observation:** An observation is an act associated with a discrete time *instant* or *period* through which a number, term or other symbol is assigned to a phenomenon.
- **Observed Property:** A property is a facet or attribute associated to an object. An observed property refers therefore to a property of the feature of interest that can be observed through the use of an instrument (or process in general).
- **Procedure:** Method, algorithm or instrument, or system of these which may be used in making an observation.

The O&M specification also presents a taxonomy for categorizing the possible types of observations. The classification proposed includes a first level where the observations are split into two groups (Cox, 2010):

- I. Observations for which the result of a single observation may be either single-valued or multi-valued, but, if there are multiple values, those values do not vary with either spatial position or time during the duration of the observation.

- II. Observations for which the result of a single observation contains multiple values that varies with spatial position, time, or both, during the duration of the observation.

For the first group, the standard proposes a further classification, centered in the characteristics of the data. This leads to a set of specialized observations that describes data having different nature.

The set of specialized observations classes suggested in the O&M specification are shown in following table (Cox, 2010):

Table 2: Specialized Observations for constant result observations.

Specialized observation class	Result type	Example*
Measurement	Measure	A measurement of "mass" (property-type) of "the seventh banana" (feature-of-interest) using the "kitchen scales" (procedure) had the result "150g".
Category Observation	Scoped Name	A category observation of the "taxon" (property-type) of "specimen 123" (feature-of-interest) by "Amy Bachrach" (procedure) had the result "Eucalyptus caesia" (from the Flora of Australia).
Count Observation	Integer	A count observation of "the number of votes cast" (property-type) at "the municipal election" (feature-of-interest) using the "electronic voting machine tally" (procedure) had the result "3542".
Truth Observation	Boolean	A truth observation of "presence" (property-type) of "intruder" (feature-of-interest) using the "CCTV" (procedure) had the result "false".
Temporal Observation	Temporal Object	A temporal observation of "duration" (property-type) for "Usain Bolt 100m dash" (feature-of-interest) using the "stop watch" (procedure) had the result "9.6s". ^a
Geometry Observation	Geometry Object	A geometry observation of "perimeter" (property-type) for "plot 987" (feature-of-interest) using the "field survey GHJ" (procedure) had the result "(... description of polygon ...)".
Complex Observation	Record	A complex observation of "major element composition" (property-type) for "specimen h8j" (feature-of-interest) using the "ICPMS" (procedure) had the result "(... array of element proportions ...)".
^a If the observedProperty of a temporal observation is 'occurrence time' then the value of the result will generally be the same as the value of the phenomenonTime. Note: The classes Measure, ScopedName, Integer, Boolean, Record are defined in ISO/TS 19103:2005. * Examples taken from O&M specification.		

Under the second category, the values can vary either with time or space. Usually, as the observation samples the target feature in a discrete way, the result is a discrete set of values sampled in time or space. This kind of results is called coverage, and due to its discrete nature, is also called *discrete coverage*. For this case the standard also proposes a set of specialized observation types depending on the possibilities of variability in time, space or both.

The following table (Table 3) shows the specializations presented in the specification.

Table 3: Specialized Observations for variable result observations.

Specialized observation class	Result type	Example*
Discrete Coverage Observation	Discrete Coverage	Is the base class of the specialized observations.
Discrete Point Coverage Observation	Discrete Point Coverage	The color of a scene varies with position. The result of an observation of the property "color" of the scene is a coverage. Each domain element is a pixel whose index allows the spatial location within the scene to be obtained.
Discrete Time Series Observation	Discrete Time Instant Coverage	An air- or water-quality monitoring station observes properties such as ozone, turbidity, etc. The instantaneous value is a scalar concentration or index value. However, the value is time-dependent. The value may be expressed as a coverage whose domain is the period of interest. This is usually described as a time series, which is a discrete time coverage.
*Examples taken from O&M abstract specification (Cox, 2010)		

Besides the existence of these types, the XML implementation of the specification allows the use of 'custom' observations types. This has importance for keeping the standard flexible enough to accept cases that might not fit this set of observations types, but at the same time can affect the interoperability of the standard, due to the use of types of observations that are not considered in the standard.

2.3 Related Work

In (Nüst, Bache, Bröring, Stasch, & Jirka, 2010) is presented a work for visualizing sensor data availability, providing different time charts for highlighting the presence

of data for different time periods. This work does not directly address the visualization of sensor data. Instead, the visualization performed is focused in showing characteristics of availability of data kept in SOS servers. In (Stephan Nebiker, 2006) is presented a project that involves a wider perspective for visualizing different types of content in a virtual globe. In this case, it is not presented an analysis including possible visualizations types that could be applied to sensor data. With the information provided in the paper is difficult to determine the possibilities offered regarding sensor data retrieval and handling for performing the visualizations.

In other works like the one presented in (Liang, Chang, Badger, & Rezel, 2010) is performed the integration of the sensor web with virtual globes. It is included some visualization types for showing the historical behavior of the sensor data through the use of time series. Nevertheless, is not explained the types of data that can be represented, neither the visualization types supported.

In (Heavner, Fatland, Hood, & Connor, 2011) is presented SEAMONSTER, a project that included the creation of a tool for visualizing sensor data. The project does not use the SOS standard. Instead, it is used the WFS (Vretanos, 2010) standard for publishing the observations. In the paper is not discussed the visualization types supported nor if this tool could be generalized for visualizing sensor information outside the scope of the project.

In general, in the papers mentioned above, is not performed an integration with analysis tools. Also is not presented a case-based analysis for achieving data visualization. In some cases the projects cover many other areas (i.e. system architecture, scalability, interaction with social networks) and the visualization of sensor data is intended for educational purposes. Therefore, the projects have lower requirements in terms of data handling, exploration and visualization.

2.4 Summary

In this chapter we have addressed the main technologies relevant to the task of visualizing sensor data. Special emphasis has been put in the SWE standards and the features related with accessing, filtering and retrieving sensor data, given the

importance that it has for selecting the appropriate data to visualize. It has been included the main types of observations and their characteristics; elements that we use as the basis of the analysis for possible visualizations types. Finally, we included some of the existing works related with sensor data visualization.

3 SENSOR DATA CLASSIFICATION FOR VISUALIZATION

In this chapter we have the purpose of exploring the different types of sensor data and suggesting a set of visualizations that can be used. In the previous chapter was mentioned the taxonomy of the sensor data, but it was centered in one observation. Besides, for visualization purposes that perspective has to be widen in order to include situations in which not one, but many observations have to be visualized. The set of visualization types proposed in this chapter are just a sample that we might use later in the implementation.

To visualize sensor data, we have analyzed and included some suitable visualization methods for the data depending it characteristics. The analysis considers elements such as cardinality, nature and dimensions of the data (describing its temporal and spatial behavior). As a start point we use the classification of observation by result type mentioned in the Section 2.2.5:

- I. Observations for which the result of a single observation may be either single-valued or multi-valued, but, if there are multiple values, those values do not vary with either spatial position or time during the duration of the observation.
- II. Observations for which the result of a single observation contains multiple values that varies with spatial position, time, or both, during the duration of the observation.

This initial classification allow us to split the observations that exhibit a static behavior regarding temporal and spatial components, from those whose values vary in time and spatial position during an observation. It has a big impact as leads to different visualization methods in order to transmit to the user the characteristics of the data. This categorization is used for analyzing the possible visual representation types that can be used in presence of one or multiple sensors.

In the following sections we suggest some possible visualization types for each case.

3.1 Data belonging to category I

Inside this category the sensor data may be coming from one or multiple sensors offering different visualization possibilities.

3.1.1 One sensor

If the data is coming from a single source (procedure), the visualization could include metadata obtained from a *DescribeSensor* operation or data coming from the result of a *GetObservation* operation. In the later case, regarding the number of observations to visualize exist the following possibilities:

- *One observation*: Visualization methods in this case could be based on just representing the observed value or, if appropriate, some categorization of the data. The categorization could be useful for doing some generalization on the values of the observations. For example, it would be useful to define value ranges and map these ranges to certain characteristics of the elements in the visualization (i.e. color or size). In this case it is necessary to consider the nature of the data. In general, it is possible to visualize this data by showing a textual representation of it.

Besides this representation would be also appropriate to use visual elements that are somehow linked to the observed value. For example, in the case of data of type Measure (see Table 2) it is possible to include some shape with attributes linked to the value to visualize. Examples of attributes of the shape that could be linked are the size, the color, or the form of the shape. In other cases such as count, or truth observations the same general technique could be applied.

- *Multiple observations* (sequence of observed values gathered in various observations): When there are multiple observations, with different sampling times, it would be expected to see somehow the evolution of them through the time. For this purpose could be used different kind of chart visualizations such as bar charts, time series charts, or difference charts (see examples in Figure 2, Figure 1, Figure 4 and Figure 3). Time series could involve different time periods in the same chart. This is helpful for analyzing the behavior of data in similar

time periods, for example, for visualizing the temperature behavior in the same month of different years.

In this case it is also possible to use animations that show the behavior of the observations values through the time. At a single instant, the animation can show a representation based on the methods proposed in the previous item, referring to one observation.

3.1.2 Multiple sensors

Having to visualize data from multiple sensors opens the possibility for a new set of visual representations. In this case, the visualizations have to favor the comparability of the provided data and the exploration of the possible relations existing in it. For example, it is desirable to show the temperature measured in two different places through the year. Another example could be to show how two different properties (i.e. temperature and humidity) are correlated in a given data set.

Regarding the cardinality of the observations, the following possibilities exist:

- *One observation:* The visualization can be done repeatedly in the same way as for a single sensor (see 3.1.1) and, if appropriate, use the same representation for data from similar observed properties, in order to ease the comparison between the visualized data. For example, it could have the same visual representation temperature data placed in two different places.
- *Multiple observations:* As mentioned before, in the presence of multiple sensor data, it is possible to create visualizations that allow analyzing potential correlations between data. Possible visualizations could use scatter plots, line charts, and time series. It is also possible to show an animation where the animated properties are linked to the values of the data. Should be considered that there is not guarantee about having values of the different sensors in a common time range, which could limit the possibility of getting meaningful charts.

Figure 2: Example of Difference Chart

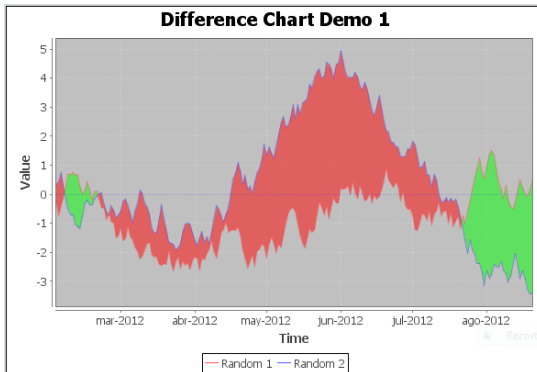


Figure 1: Example of Scatter Plot Chart

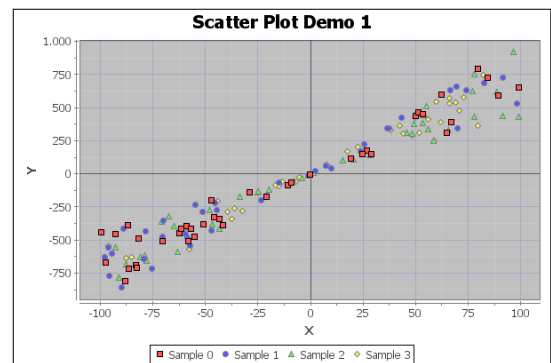


Figure 4: Time series chart

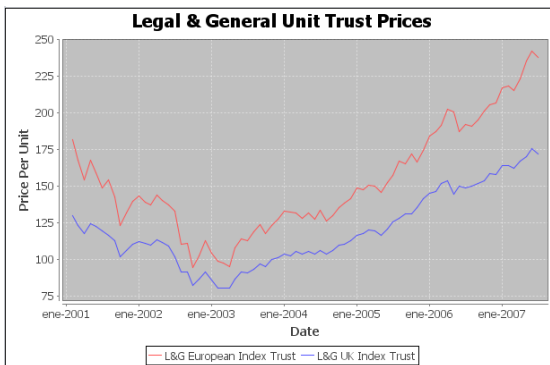
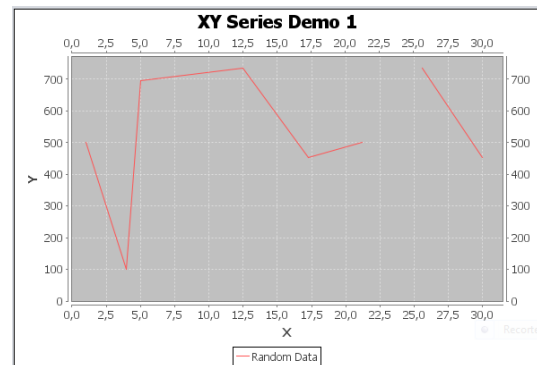


Figure 3: Example of XY Chart



3.2 Data belonging to category II

Under this category we can classify data depending on its spatial and temporal variability. The visualization of multiple observations can use animations using the visualization types proposed for one observation (presented below).

3.2.1 Data that varies its spatial position

In this category it is necessary to visualize data by using some method that shows the values distributed in the space. Some possible visualization methods could be contour lines, dot distribution¹⁵ maps-like representation, and analytic surfaces (surfaces where the height at given point depends on the value observed in that position, see Figure 5).

¹⁵ http://en.wikipedia.org/wiki/Dot_distribution_map

Contour lines¹⁶ are useful to compare the different values belonging to the observation, and to visually identify zones where the observation exhibit homogeneous values. Through the use of dot distribution map, is possible to show the values using categories defined by the ranges of them. The categories can be linked to different colors that can be assigned to the dots in the representation.

3.2.2 Data that varies with time

For visualizing data in this category, can be used animations showing the values along the time (scaled) of the observation. It is also appropriate to use time series charts showing the values during the time of the observation.

3.2.3 Data that varies in space and time

This case can be seen as a conjunction of the previous two cases. A possible visualization method is the dynamic analytic surfaces, analytic surfaces animated so that the notion of time is incorporated through the animation.

Figure 5: Examples of Analytic Surfaces.

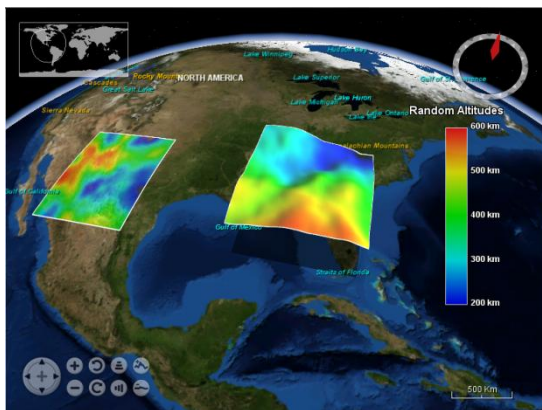
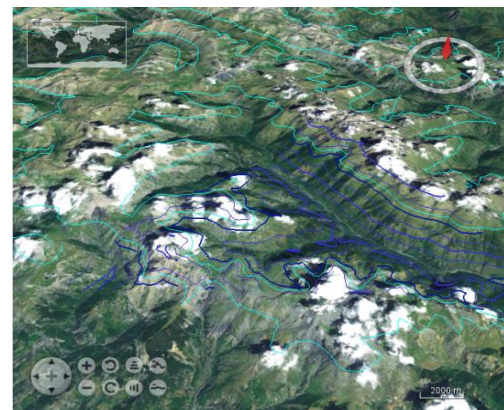


Figure 6 : Example of Contour Lines



3.3 Other types

Besides the previous two categories, it is necessary to introduce a further generalization. It would be the case in which the position of the observed value or

¹⁶ http://en.wikipedia.org/wiki/Contour_line

values varies from one observation to another, despite their spatio-temporal behavior during the observation. This situation is more proper of mobile sensors.

The visualization methods in this case can include paths, for visualizing the spatial distribution of the different observations. Appropriate visualization techniques have to show the order in which the observations are performed in time, and for visualizing each observation can be used the methods mentioned previously (suggested in section 3.1 and 3.2).

So far we have addressed visualization methods that can be derived from the direct use of sensor data, without involving any further data analysis. The use of analytical tools can help to include results of analysis as part of the visualizations. Such integration can ease data exploration and enrich the information provided in the visualizations.

The analytical tools, through the use of different geo-algorithms over sensor data, can result in different types of content. The results of applying different algorithms can be categorized as follows:

- Geometries: The geometries can be of different types, lines, polygons and points. An example of such type of result is buffers. A buffer can be used to create a hypothetical influence zone of a certain data measured in a given location. The values measured can be linked to the ratio of the buffer with its center of the sample position.
- Textual results: Some geo-algorithms calculations have their outputs in form of text. For example, basic statistics about the data could give a text containing the calculations results.
- Tables: Some algorithms can give as output a tabular content with the result of the calculations.
- Raster data: Some analytical tools can generate raster data or georeferenced images. For example, interpolation techniques like some of the Kriging¹⁷ family have raster outputs.
- Charts: Charts can be a form of result in some of the geo-algorithms.

¹⁷ See <http://en.wikipedia.org/wiki/Kriging>

All this kind of outputs created by geo-algorithms can be visualized and integrated with the different visual elements provided by the virtual globes.

3.4 Summary

In this chapter we have explored a set of visualizations methods that can be suitable for different types of sensor data. The visualization techniques proposed rely on a categorization that takes into account the cardinality of observations and their spatio-temporal behavior. Our aim with this classification is to have a basis for implementing different kind of visualization methods in the prototype tool presented in this work. The set of visualization techniques proposed does not have the intention to be complete but only to guide the development of the prototype.

Most of the emphasis has been put in the visualization methods for data whose values that do not vary in space and time during the observation, as in our opinion is the most commonly published. In the analysis has been also considered types of outputs that can be obtained from the application of spatial analysis methods.

4 PROTOTYPE DESCRIPTION

In this chapter we present a prototype application for visualizing sensor data. The following sections will expose the requirements, the general structure and the features included in the prototype.

4.1 General Description

For designing the tool we started with a set of requirements:

- The tool should be generic, so that the sensor data can be retrieved from different SOS compliant servers.
- The prototype has to ease the interaction with SOS servers for accessing and retrieving sensor data.
- It is necessary to provide functionality for handling the sensor data to be visualized. It has to be provided the possibility of creating and mixing data from different sources.
- The prototype has to include a set of visualization methods that can be applied directly to sensor data.
- The integration with SEXTANTE has to be performed for supporting analysis and data processing for further representation of the analysis results.
- Use the NASA World Wind for Java for placing the visual representations and offer ways for controlling the visualized content.

The implemented tool uses the virtual globe NASA World Wind for Java to visualize sensor data. At high level, it is client application that can connect and interact with SOS servers. The main visualizations types offered by the prototype tool cover mainly the ones listed in the previous chapter under the category I (see 3.1). Our aim is to create a general tool, with the possibility of connecting and retrieving data from different SOS servers and performing a set of visualizations using this data.

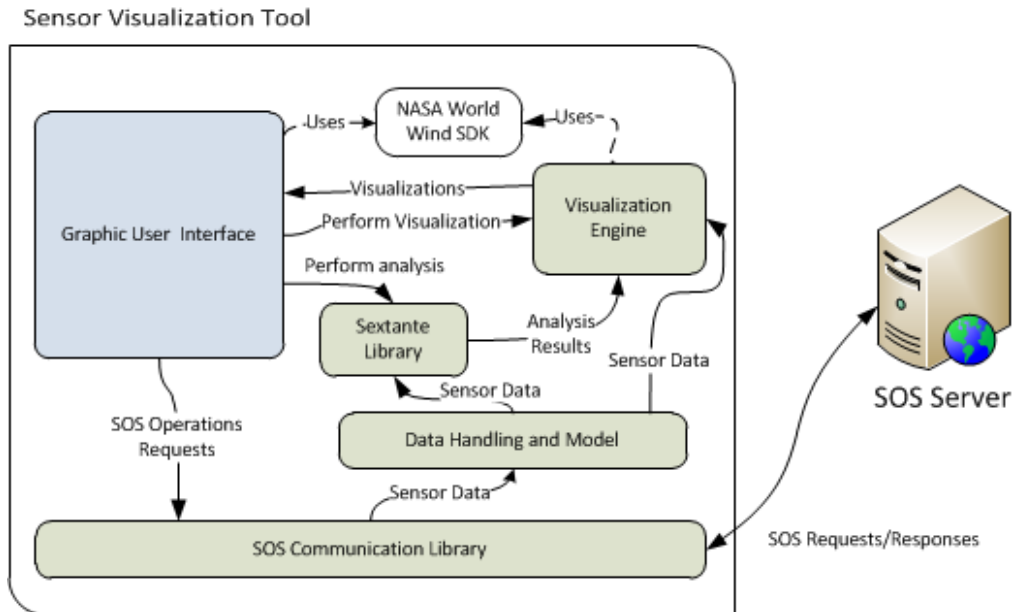
The tool is implemented in Java using the Eclipse RCP framework¹⁸. The Eclipse Rich Client Platform (RCP) framework has a set of features that helps in creating a dynamic customizable user interface. Such customization is important in this tool as

¹⁸ http://wiki.eclipse.org/index.php/Rich_Client_Platform

allows hiding interface elements that are not needed while exploring the performed visualization.

A general depiction of the main components of the tool can be seen in Figure 7:

Figure 7: Main components of the prototype.



The main components of the tool are the NASA World Wind SDK, mainly used for visualizing the data and interaction with the visualization elements. The communication library, which interacts with the SOS servers for accessing and retrieving the sensor data, and the SEXTANTE library for providing the analytical tools for performing analysis over the sensor data.

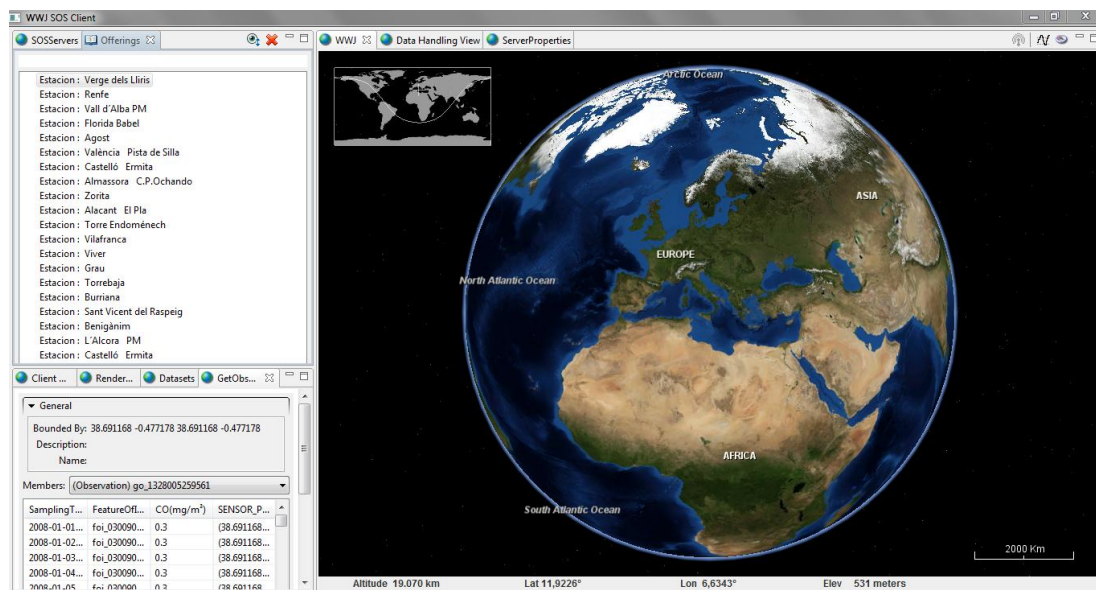
The user interface is organized in different views that have specific roles in the application. The views provide functionality for performing different tasks:

- *Servers View*: Contains the functionality for connecting to SOS Servers and keeps a list with the connected servers.
- *Server Properties View*: This view shows the information regarding the currently selected server in the Servers View.
- *Offerings View*: Contains the offerings existing in the currently selected server.

- *Messages View*: Shows the content of the request and responses exchanged with the SOS server.
- *Globe View*: Contains the NASA World Wind Virtual Goble embedded, used for the visualization of sensor data.
- *Data Handling View*: This view allows performing operations over the datasets retrieved from the SOS servers in order to create new datasets.
- *Datasets View*: Shows the general structure of the existing datasets in the system. This could include the datasets retrieved from the SOS servers and the datasets that might be created in the Data Handling View.
- *Rendered Objects View*: This view includes functionality that allows controlling the visible state of the rendered objects.

Figure 8 shows the composition of the user interface of the prototype.

Figure 8: General composition of the prototype.



The tool tries to ease the tasks of accessing, retrieving and visualizing sensor data by providing different wizards. This feature allows guiding the user through the configuration of parameters and the selection of the visualization type that might be complex otherwise. In the following sections we offer more details about the wizards included in the tool.

4.1.1 Communication Library

The prototype tool communicates with the sensor services using an interface provided by a light-weight communication library presented in (Tamayo, Viciano, Granell, & Huerta, 2011). The library, implemented in Java, provides the access to the SOS servers and hides the complexity of encoding and decoding the messages exchanged with them.

To interact with SOS servers, only operations included in the *core profile* (section 2.2.1) are supported. These operations are enough to achieve the purpose of addressing the generality of the SOS compliant servers, as they are the only mandatory and the functionality provided in this profile enables the consumption of sensor data. In addition, the library offers an API that ease configuring the parameters for executing these operations. It is supported the configuration of filters for *GetObservation* operation. This feature is fundamental as it helps in the process of sensor data selection.

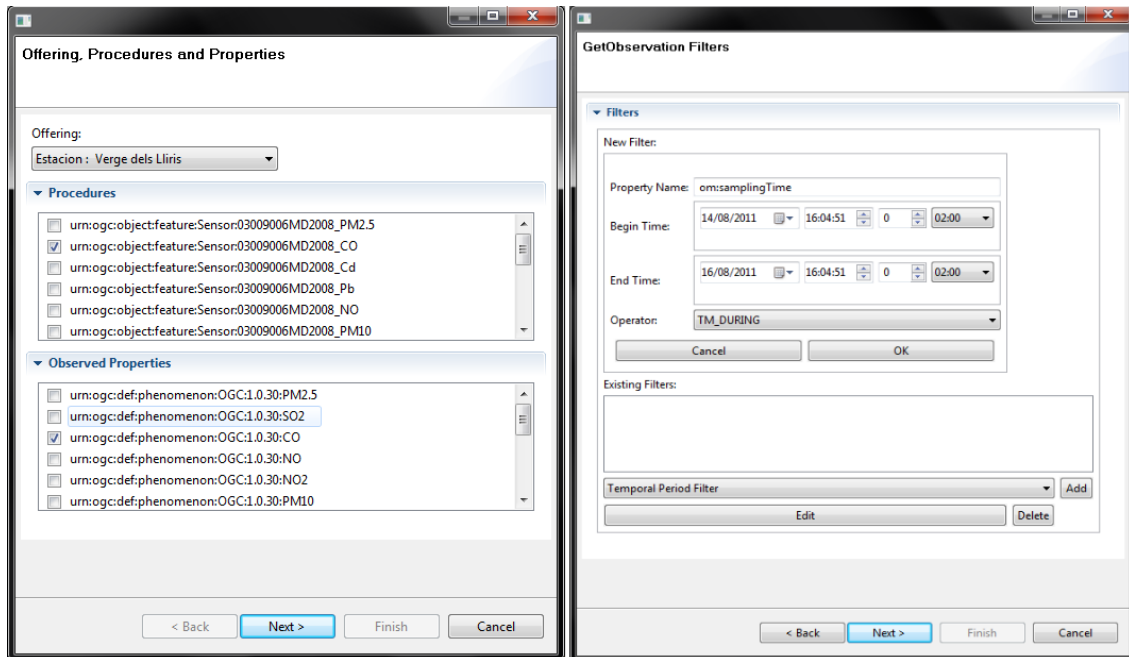
Regarding the sensor data retrieved, the library includes an object model that represents the data encoded in O&M (see 2.2.5) and to a less extent to SensorML (see 2.2.4).

4.1.2 GetObservation operation support

The *GetObservation* operation has some complexities related to the parameters necessary to invoke it. As we discussed in the section 2.2.5 most of the parameters depend on the results of the *GetCapabilities* operation, which advertizes the features supported by the SOS servers and the sensor data (*Offerings*) hosted by them. The tool provides assistance for executing this operation by offering a wizard that helps in the configuration of the parameters and the different types of filters.

Figure 9 shows two of the pages of the wizard. The page on the left allows the selection of the offering, procedures and the observed properties for retrieving the sensor data. On the right, the wizard page shows the interface for configuring the filters. This page provides an easy way for the creation and edition of the different types of filters mentioned in 2.2.5.

Figure 9: GetObservation wizard pages.



4.1.3 Data Handling and presentation

For visualization purposes, it is important to be able to select the data, filter and possibly create new data by mixing information from different sources. When executing a *GetObservation* operation, the server response contains the data separated in structures (called members). The separation in members is done according to the different observed properties selected during the configuration of the request.

The tool keeps the data retrieved from the SOS servers in form of datasets associated to the request. Each dataset is composed by three main fields, the sampling time of the observation, the observed value, and the feature of interest. After retrieving the dataset we have incorporated an extra field for including the sampling position.

The tool includes two different ways for presenting the datasets reused in different views. One of the components used for presenting the datasets shows a structural representation of it in which is possible to see and select its fields (see Figure 10). The other dataset component is used in the *GetObservation Result View* (see Figure 11). In this case the data resulting from a *GetObservation* operation can be visualized and inspected.

Figure 10: Structural View of the datasets.

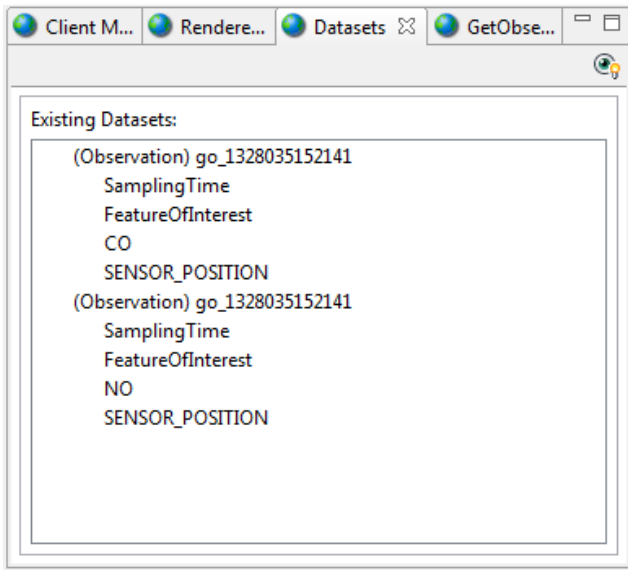
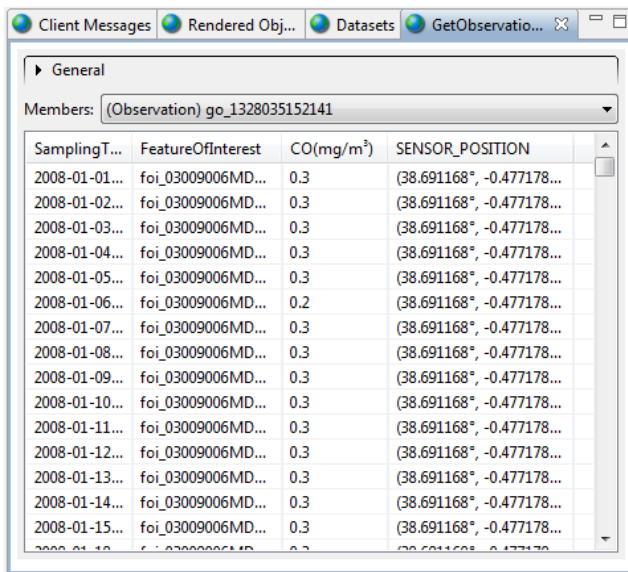
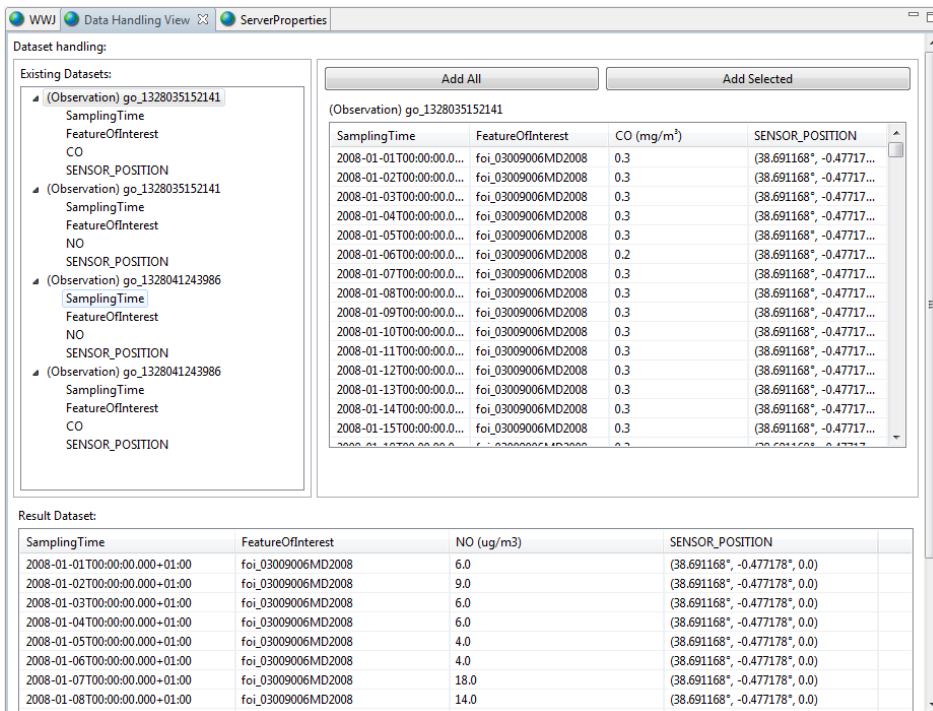


Figure 11: GetObservation results view.



Besides these features, the tool incorporates a view (Figure 12) to select data from the obtained datasets and combine them for obtaining new datasets. In the Figure 12 there is data from different *GetObservation* operations that can be mixed for creating new datasets. This is particularly important when is necessary to have values sampled in different locations and the user wants to perform and visualize an analysis involving such data.

Figure 12: Dataset Handling View.



4.1.4 Integration with SEXTANTE library.

For favoring data exploration, we decided to include analysis capabilities to the tool. For achieving this purpose we integrated the library SEXTANTE with the prototype. This open source library contains a collection of more than two hundred algorithms that provide different analysis capabilities.

SEXTANTE is generic in the sense that provides functionality that allows adapting the object model of the application using the library to the inputs of the algorithms offered in it. Therefore, for accessing to this rich collection of algorithms is enough to write a set of adapters to feed any algorithm with the necessary data. Additionally, SEXTANTE provides a mechanism for including new algorithms to the library without having to recompile or modify the library directly (Olaya, 2007). This feature is very desirable for incorporating custom algorithms with context specific purposes.

Other powerful feature of SEXTANTE is the integration with R¹⁹, a widely used tool with endless statistical algorithms. The library also provides visual tools that can be

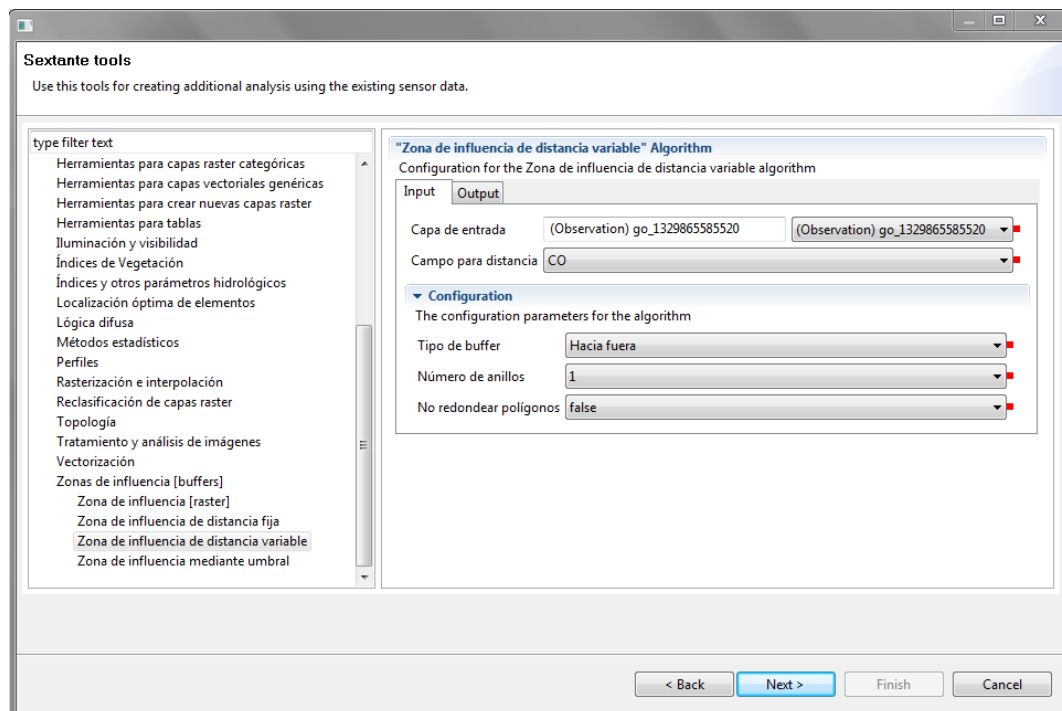
¹⁹ The R Project: www.r-project.org/

integrated with custom software using the Swing²⁰ graphic user interface library. The visual tools give access to the algorithms and enables fine control over the parameters provided to them through the user interface.

Unfortunately the platform Eclipse RCP uses a different graphic library (SWT²¹) that is not compatible with Swing. Due to this limitation, it was necessary to use and adapt other components for providing the user interface access to the algorithms of the library. The component library used was an UDig²² implementation of the graphic tools of SEXTANE for Eclipse RCP. It has big dependencies with the UDig framework so we had to perform substantial modifications to the code in order to reuse it.

The user interface for accessing the functionality of SEXTANTE provided by the tool has two main areas: the area for selection of the algorithm and the area for the parameters of the algorithm selected (Figure 13).

Figure 13: SEXTANTE library user interface.



²⁰ Swing <http://java.sun.com/products/jfc/tsc/articles/architecture/>

²¹ Standard Widget Toolkit <http://www.eclipse.org/swt/>

²² UDig (GIS Framework for Eclipse)

<http://udig.refractions.net/>

SEXTANTE defines a set of types of data inputs that can be used by the underlying algorithm library. The types of data inputs are summarized in Table 4:

Table 4: SEXTANTE input parameter types.

Input type	Description
Raster Layer	Provides the library with raster data used in many of the algorithm provided.
Vector Layer	Represent a table – like data structure containing a field with a geometry and a set of fields containing the data associated to that geometry.
Table	Is a tabular data structure with a set of fields or attributes.

The implementation of the adapters performed in the visualization tool includes only the Vector Layer and the Table input types. This is conditioned by the type of sensor data handled in the application. As the tool does not support the consumption of image data is not possible to provide this type of data to the SEXTANTE library.

For implementing the adapters for the Vector layer type, the geometry field provided to SEXTANTE refers to the sample position. The rest of the fields associated to the geometry field are the sampling time, the feature of interest and the observed value field. For the Table input type the fields provided are only the sampling time, the feature of interest, and the observed value field.

The integration with SEXTANTE was mainly for visualization purposes, so at this stage of the project, some functionality of SEXTANTE related data handling is not leveraged.

The results of the analysis performed with SEXTANTE can have different types. They fall in five main categories, for which the tool produces a different visual representation. The types of outputs are:

- Raster data: Many algorithms produce raster data as output. For example, interpolation algorithms.

- Textual data: Many types of calculations, as for example basic statistics produce this kind of output, in HTML format.
- Charts: Some algorithms produce charts as outputs. The charts produced are objects of the JFreeChart²³ chart library.
- Tables: Tabular data produced by the algorithms. For example, some of the algorithms included in the library produce some modifications on the tables provided as input.
- Vector layers: This type of output is basically tabular data containing a geometry field associated to a set of fields containing other sort of data.

These types of outputs are represented in the globe in different ways. The vector layers produce visualizations in which the geometries are represented in the virtual globe. The rest of the data associated to the geometry is represented as a textual content, through the use of balloons. The balloons are then placed in the centroid of the geometry to indicate the relation existing between data and its geometry. Other types of outputs such as charts, tables, and textual data are placed in balloons in the globe. For determining the position of the balloons in the globe the position field existing in the dataset is used.

4.2 Visualizations

The visualizations supported in the tool fall in two main groups: data visualizations and non data related visualizations. The first group of visualizations is based on data obtained from sensors systems and it is intended to visualize information for facilitating data exploration and analysis. The second group is related to the representation of general information about the sensors and observations, such as its spatial distribution and other metadata of interest.

For creating the visualizations related with sensor data, it is necessary the intervention of the user in the process of selection of the data and the configuration of the visualization. The process of data selection occurs in various stages of the application, starting by the execution of the *GetObservation* operation, where it is possible to filter the data using different criteria. A further stage of the process of

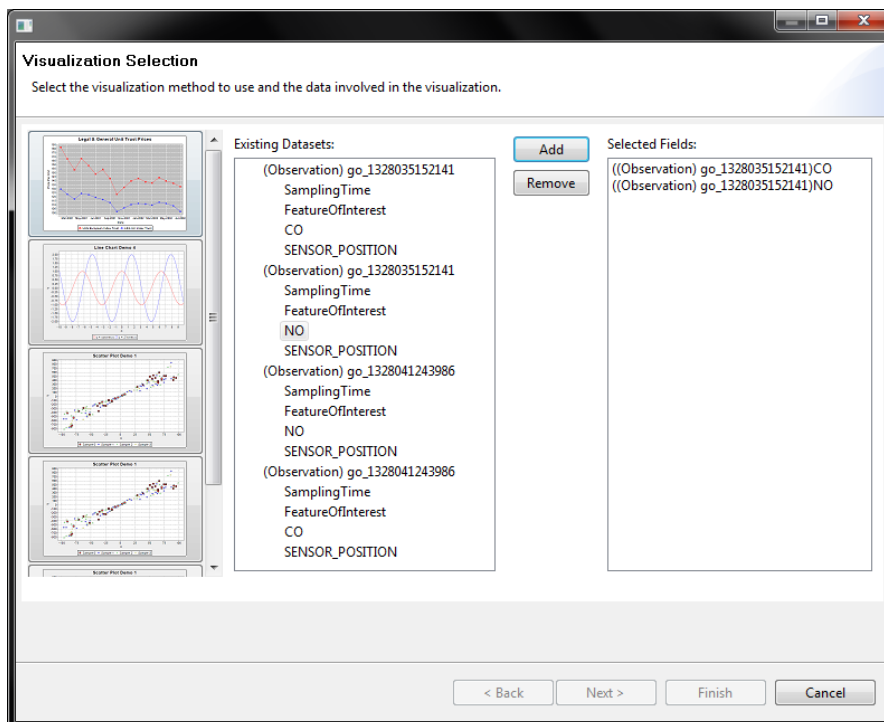
²³ <http://www.jfree.org/jfreechart/>

data selection is supported by the data handling capabilities included in the tool. Finally, for the visualization the user can select the data to be visualized from the list of existing datasets.

The process of data visualization is assisted in the tool by providing a wizard where is possible to select from different types of visualizations and perform some configuration before including it in the globe. The visualization methods include three types of different charts and one animation where is possible to see the progress of the observations along the time. It is also included a more general set of visualizations relying in the application of analysis algorithms using SEXTANTE library.

In Figure 14 can be seen the first page of the wizard for assisting in the selection and configuration of the visualization. The wizard has controls for choosing the type of visualization and the data to be included in the representation.

Figure 14: Visualization wizard.



4.2.1 Chart visualizations

The chart visualizations are a good tool that allows presenting the information in a very concise way. In the tool we have included three types of charts, time series charts, scatter plot charts, and XY charts.

The charts produced can be customized before visualizing so that elements such as the title, font, background color, etc. can be modified according to the needs of the user. It is also possible to zoom in or out in the chart to focus in a particular area of it before its inclusion the globe. The configuration page is part of the visualization wizard (see Figure 15).

The time series charts are particularly useful in this context due to the characteristics of the sensor data. Usually there are big amounts of historical data and it is necessary in many cases to visualize the trends and the relation with other data in the same (or similar) time period. The Figure 16 shows the visualization of two different observed properties (NO and CO) in the period of January 2008 - January 2009.

Figure 15: Chart Preferences page.

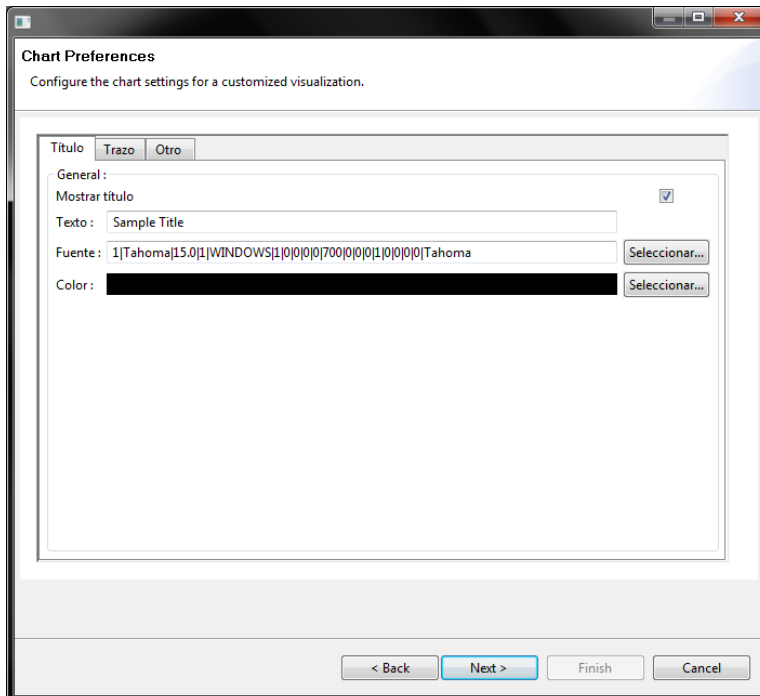
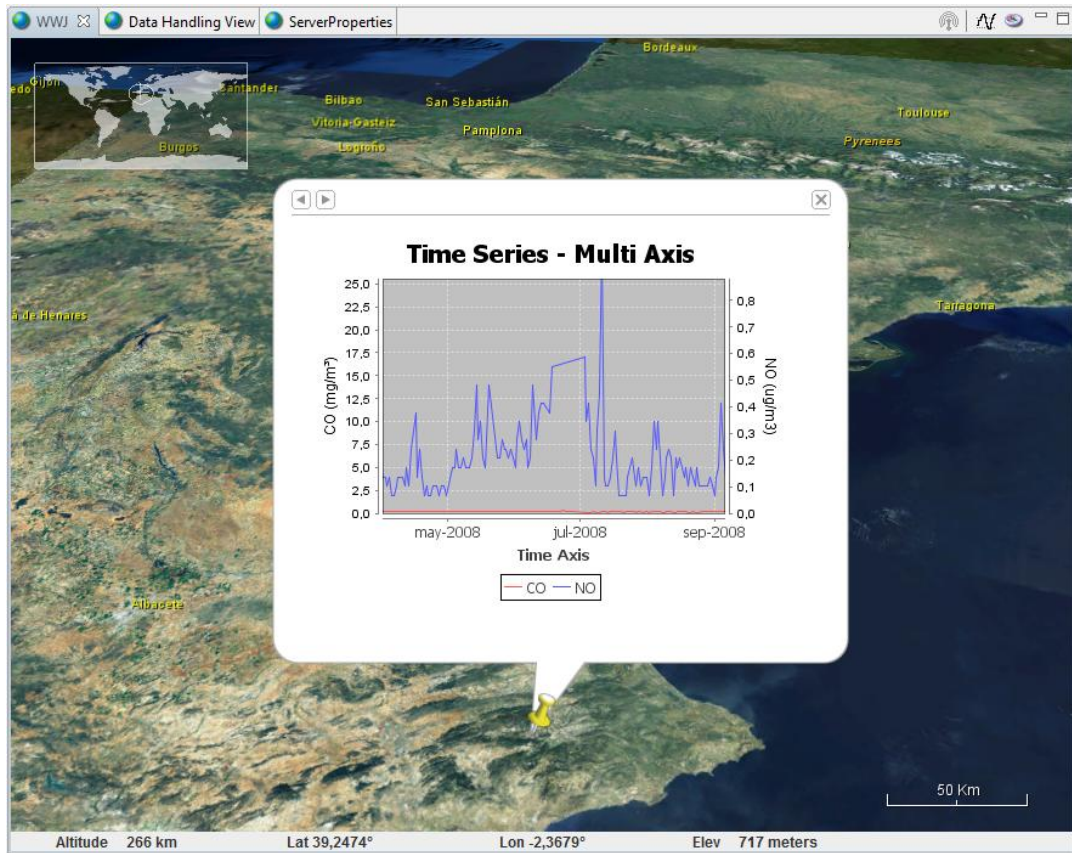
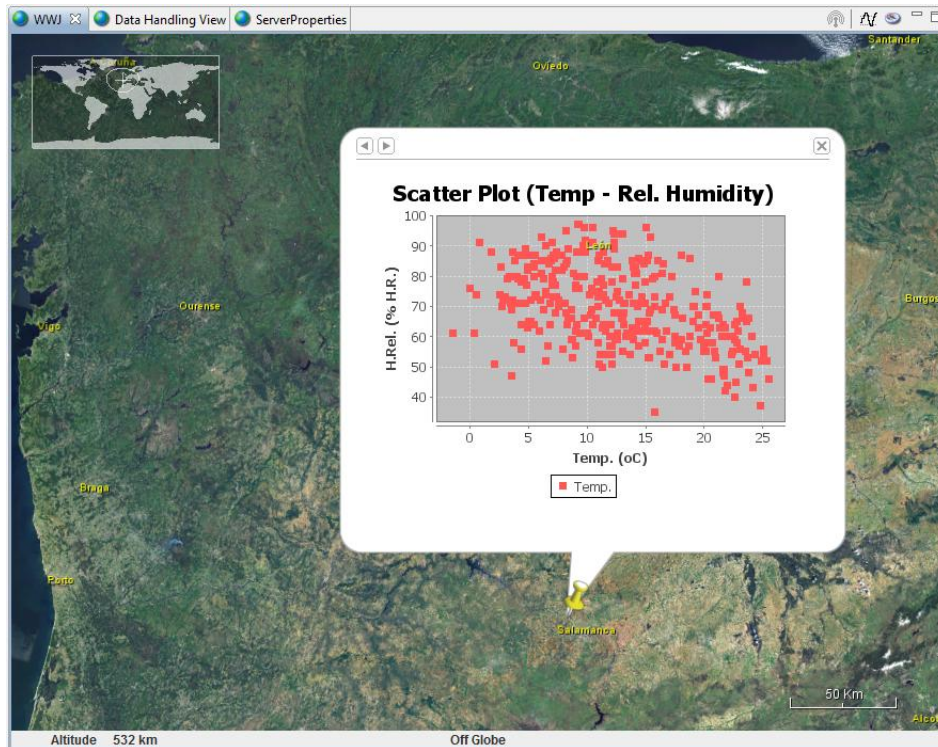


Figure 16: Time series charts with multi-axis support.



The second type of charts supported is the scatter plot charts. Scatter plot charts are helpful for discovering correlations in data. For using this kind of charts, the user has to be aware of the limitations of applying this method for inferring correlations. In general the variables tested should be observed in the same location. The tool does not advise the user about this kind of issues as it uses a knowledge that is difficult to incorporate. Figure 17 shows a scatter plot for temperature and humidity.

Figure 17: Scatter Plot for Temperature and Humidity.



4.2.2 Animation of observations.

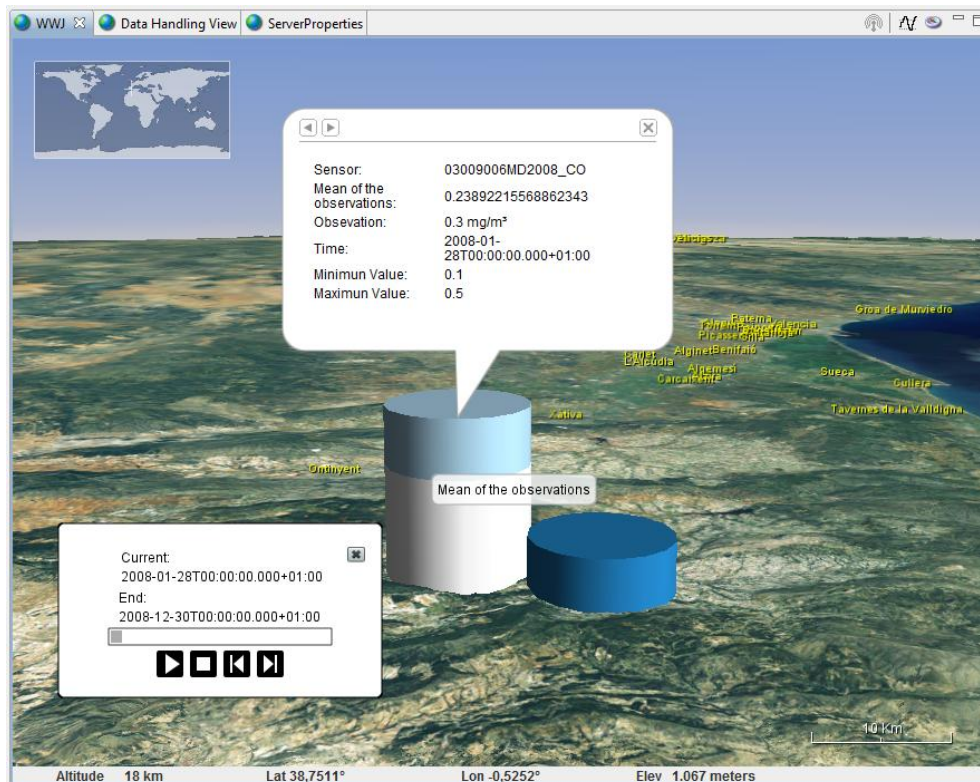
Another type of visualization included in the tool shows an animation of the observed values of one or more observed properties. This visualization uses different types of volumetric shapes for representing different observed properties. The implementation allows linking the observed value to properties of the shape such as the height or the radius.

The animation shows the behavior of the observations and includes elements for easing the comparison between them. For example, the current implementation shows the mean, maximum, and minimum of the observed values. Therefore, is easy to see the relation of the observed value at a certain instant of the animation with the mean of all the observations animated.

When animating different observed properties in the same location, the animation orders them so the visualized elements do not overlap each other.

Navigation controls for controlling the progress of the animation has been included. The controls are placed in an annotation in the virtual globe for showing the time of the observation visualized with the possibility of pausing, stopping and moving forward and backward. Figure 18 shows the animation of two observed properties in the same location. The information included associated to the visualization (see the balloon) shows relevant values, the sampling time and the sensor that performed the observation.

Figure 18: Animation of one observed property.



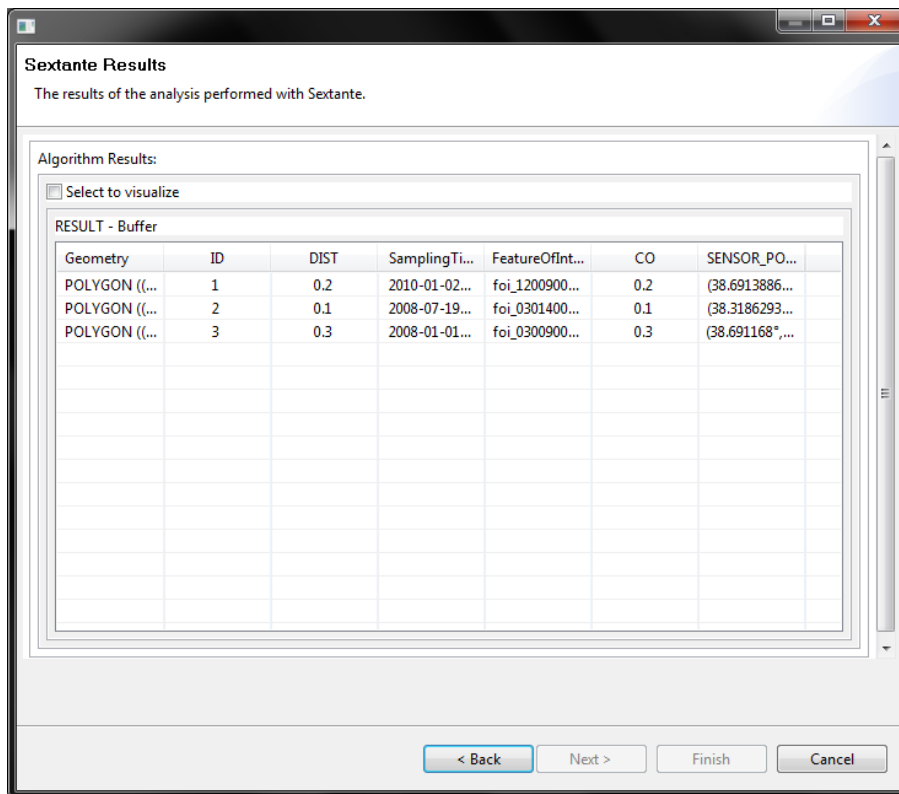
4.2.3 Visualization of SEXTANTE results.

As we mentioned in the section 4.1.4, the tool have been integrated with the spatial analysis library SEXTANTE. Trough this integration the tool is able to feed the SEXTANTE's algorithms with sensor data. This feature enables to create visualizations derived from the analysis performed with SEXTANTE.

The equivalence of the different outputs with the visual elements derived from the results and later represented in the virtual globe has been discussed in section 4.1.4. The algorithms in SEXTANTE can give several outputs to one analysis. Some of

them might not be interesting for the users, so should not be visualized. For solving this issue, the tool offers the possibility of selecting which of the outputs are going to be included in the visualization. Figure 19 shows the wizard page presented to the user for in selecting the outputs to be visualized.

Figure 19: SEXTANTE output selection.



The application of SEXTANTE algorithms depends on the purpose of the user. As an example, the Figure 20 shows a histogram of the observed values of the observations. This helps in recognizing which values are more common the dataset, which could be necessary to know in certain scenarios.

Another example is the visualization of basic statistics about the observations. Figure 21 shows the results of applying such algorithm on a group of observations.

Figure 20: Histogram produced by SEXTANTE.

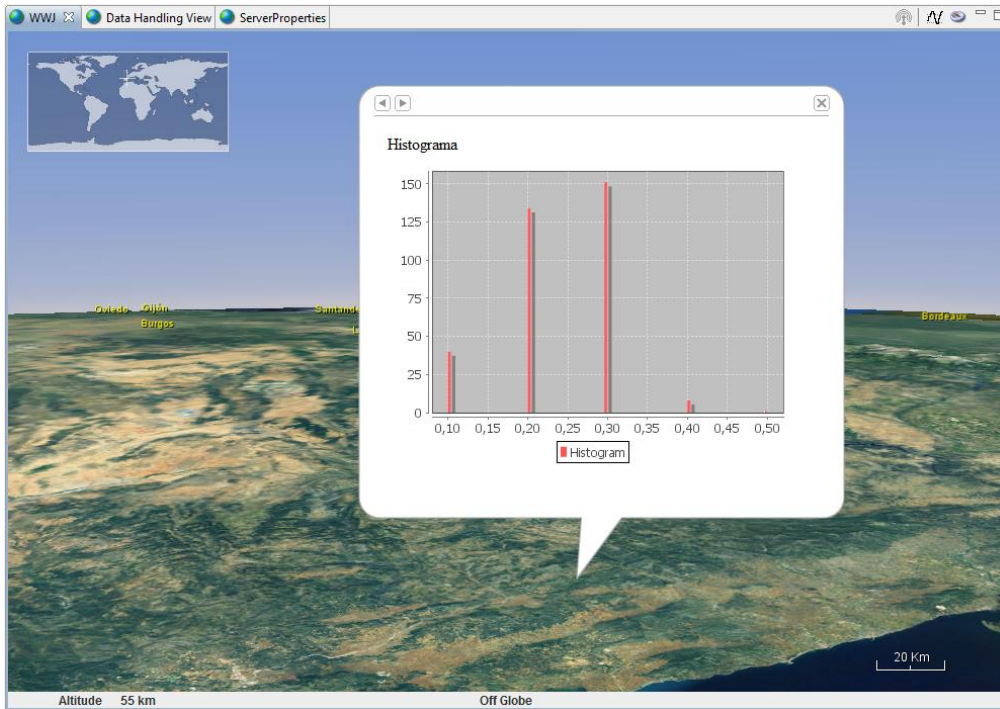
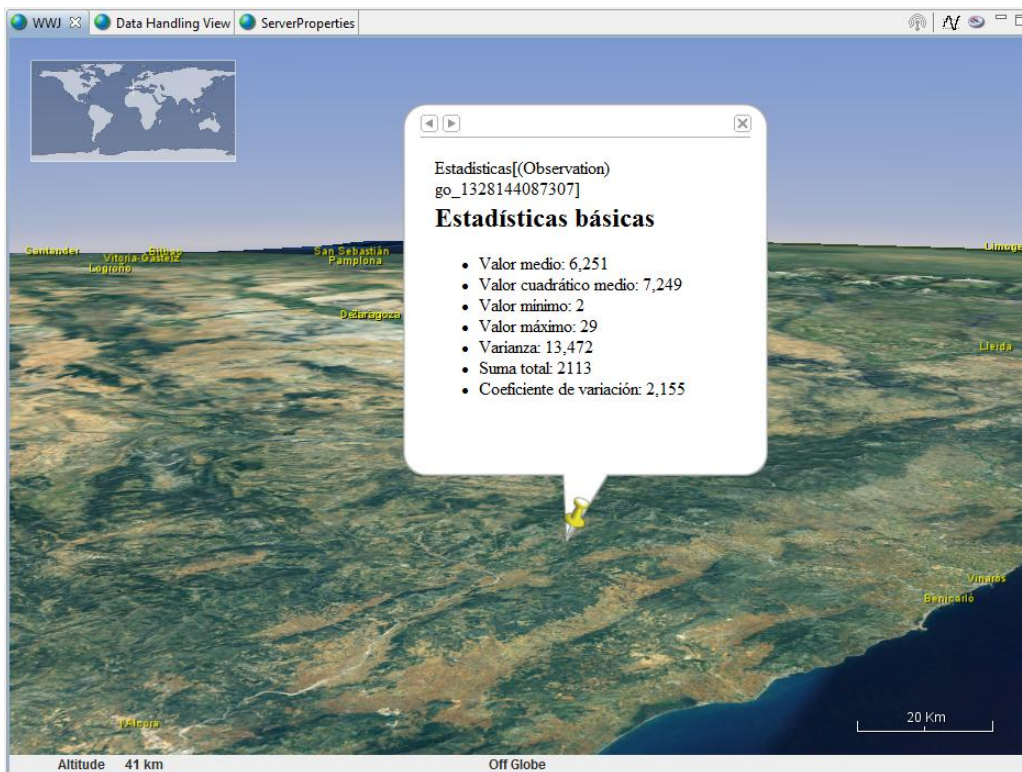


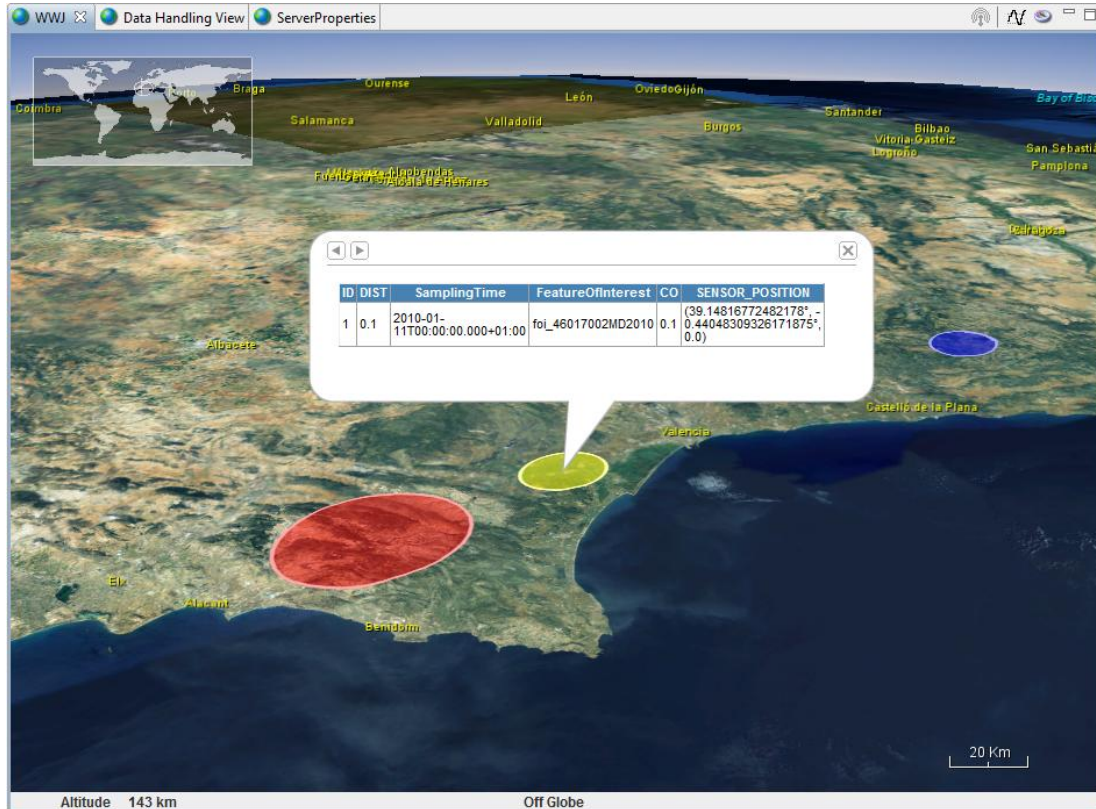
Figure 21: Basic statistics created with SEXTANTE.



Additionally, a possible basic application could be the creation of buffers based on observations in different locations. In Figure 22 are visualized buffers in which the

distance of the buffers is linked to the observed values. In the figure, the buffers are exaggerated for visualization purposes.

Figure 22: Buffers created with SEXTANTE.

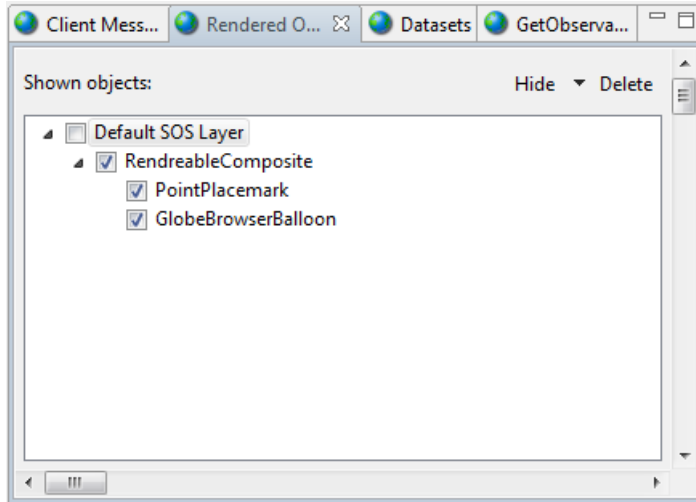


The visualizations presented using SEXTANTE are just some of the possible applications, but many other visualizations and context specific analysis can be performed by extending the SEXTANTE library.

4.2.4 Visualization Control

Having many data visualizations the virtual globe can result in cluttering. For addressing this issue the tool includes a view for controlling the visualized elements. This view shows the structure of the visualized elements for controlling their visible state in the globe. It is possible to hide or show classes of elements, for example to hide all the balloons. Additionally, this view allows the user to remove permanently the visualized elements from the layer of the virtual globe. The view is shown in Figure 23.

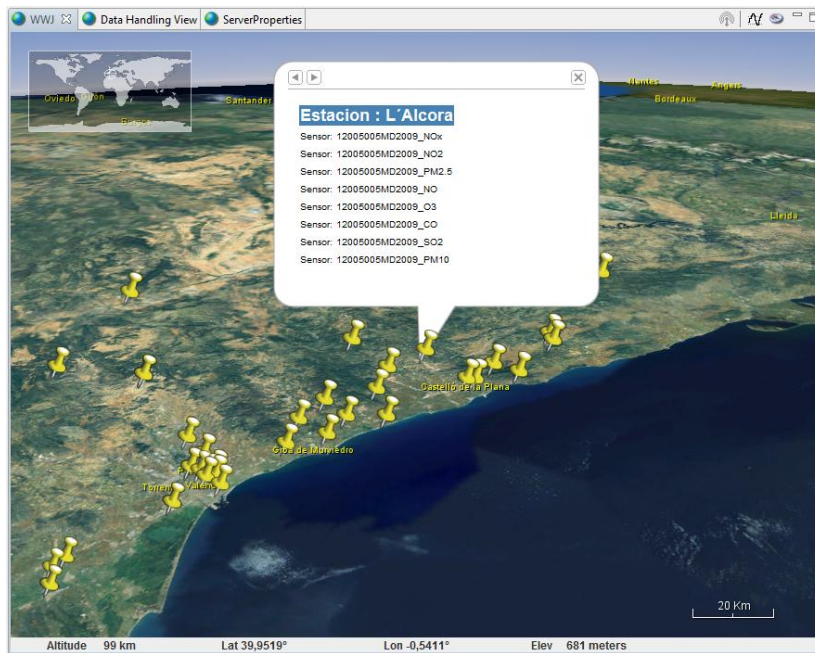
Figure 23: Visualization Control View.



4.2.5 Non-data related visualizations

As mentioned in the beginning of this chapter, the tool also supports visualizing other aspects related with the sensor web. It has been included a visualization for showing the spatial distribution of the observations advertized by a SOS server. The visualization offers basic information about the sensors (procedures) existing in the different offerings. Figure 24 shows some of the offerings advertized by a SOS server.

Figure 24: Visualization of the offerings advertized by a SOS server.



4.3 Summary

In this chapter we have provided a description of the prototype tool implemented for visualizing sensor data. It has been exposed the main features of the tool regarding sensor data access and retrieval, data handling, visualization and analysis support. Regarding the analysis support for sensor data, it has been presented how the tool performs the visualization of the different types of SEXTANTE outputs. Examples of some of the visualizations supported have been included, with a brief explanation of their contribution to the goal of helping data exploration.

5 CONCLUSIONS AND FUTURE WORK

In this work we have presented a prototype application for visualizing sensor data published using SWE standards. The tool ease the tasks of data selection and handling by providing user interface elements for achieving complex tasks such as data filtering, selection and composition.

For offering different visualization methods, the tool relies in a classification of sensor data and proposes a set of possible representations that could be suitable in each case. It incorporates visualizations for common sensor data types enabling tasks of data exploration, comparison and relations discovery. The visualization methods provided allow revealing several aspects of the data and it relations, such as time behavior and correlations. As part of the visualizations have been also included methods that show the spatial distribution of the offerings advertized by SOS servers.

The integration with the spatial analysis library SEXTANTE widens the possibilities for exploring and visualizing data as it equips the tool with a collection of algorithms with a wide range of functionalities. Such integration is even more relevant as it can include context specific algorithms for addressing unforeseen problems and integration with powerful tools as the R project.

All the features included in the tool meet the requirements established in its design and contribute to meet the objectives of this work, even though it is not exempt of some limitations.

5.1 Limitations

Currently the tool has some limitations. At the current stage, the integration with SEXTANTE does not support raster data, this is important for enabling the use of a set of algorithms. Some work has been done to incorporate this functionality but still needs refinements. The inclusion of this feature is especially important once enabled the consumption of sensor image data, currently not supported in the communication library.

Due to some limitations in the communication library (see 4.1.1), the tool does not support accessing all the metadata available regarding the sensor systems. This is a

source of data that might be of interest, given its relevance for explaining the circumstances in which the data was acquired and its quality. Knowledge about the sensors and other information about the data could help in tasks like suggesting possible visualizations. It is possible to avoid these limitations by integrating new versions of the library that already include the missing functionality.

5.2 Future Developments

For the future, many enhancements can be done to the prototype. Regarding the visualizations types, the tool should cover observation types whose values can vary in time and space. Besides, the integration of new visualizations that might be of interest can improve the functionality of the tool and adapt it to other needs. In terms of data handling, the interface should be enhanced for supporting other operations over the data besides the selection and composition. For example, operations of joining datasets are implemented but are not included in the user interface. The current implementation does not address appropriately the issues related with handling big data volumes, an aspect that should be improved to enhance the usability of the tool.

The inclusion of new functionality such as integration of data from other sources (i.e. maps services, shape files), would enhance the visualizations offered. Many other improvements need to be done related to the user interface to improve the user interaction with the tool. For example, the interaction with the elements visualized in the globe can be improved for executing actions related with them. The visualizations types' customization is an aspect that has to be also improved. The tool enables some level of customization but there are situations (as in multi-axis charts and animations) in which it is not completely effective. It is necessary to work more in this sense as in some cases the underlying implementation is prepared for providing high level of customization, but there is not access to this functionality through the user interface.

To improve sensor data access, future works can be related with the integration of the tool with catalogue services, so that the SOS services can be search and selected without needing a previous knowledge of the SOS services addresses.

Other aspect in which the project has to be developed is the ability of sharing the resulting visualization. A possible solution for this is the implementation of mechanisms for exporting the visualizations to KML documents. WWJ already include functionality for exporting to KML some of the visual elements, so this task can be achieved with relatively small amount of effort.

REFERENCES

- Andrienko, G., Andrienko, N., Jankowski, P., Keim, D., Kraak, M. J., MacEachren, A., et al. (2007). Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of Geographical Information Science* , 839-857.
- Aurambout, J.-P., & Pettit, C. (2008). Digital Globes: Gates to the Digital Earth, . *Digital Earth Summit on Geoinformatics* , 233.
- Bailey, J. E., & Chen, A. (2011). The role of Virtual Globes in geoscience. *Computers & Geosciences* , 1-2.
- Boots, M. (2007). *OpenGIS Sensor Model Language (SensorML)*. OpenGIS® Publicly Available Standard, Open Geospatial Consortium Inc.
- Botts, M., Percivall, G., Reed, C., & Davidson, J. (2007). *OGC Sensor Web Enablement: Overview*. OGC. Open Geospatial Consortium, Inc.
- CIA. (1999). *Central Intelligence Agency*. Retrieved 01 25, 2012, from <https://www.cia.gov/library/publications/the-world-factbook/index.html>
- Cox, S. (2010). *Geographic Information: Observations and Measurements*. *OGC Abstract Specification Topic 20*. OpenGIS® Abstract Specification, Open Geospatial Consortium Inc.
- Cox, S. (2011). *Observations and Measurements - XML Implementation*. OpenGIS Implementation standard, Open Geospatial Consortium.
- Craglia, M., Goodchild, M. F., Annoni, A., Camara, G., Gould, M., Kuhn, W., et al. (2008). Next-Generation Digital Earth. *International Journal of Spatial Data Infrastructures Research* . , 3 , 146-1.
- Dapple*. (2012). Retrieved 01 26, 2012, from <http://dapple.geosoft.com/>
- Echterhoff, J. (2011). *OpenGIS® SWE Service Model*. OpenGIS® Standard, Open Geospatial Consortium.

Havens, S. (2007). *OpenGIS® Transducer Markup Language (TML) Implementation Specification*. OpenGIS® Publicly Available Standard, Open Geospatial Consortium Inc.

Heavner, M., Fatland, D., Hood, E., & Connor, C. (2011). SEAMONSTER: A demonstration sensor web operating in virtual globes. *Computers & Geosciences*, Volume 37 (1), 93-99.

IAGT. (2012). *The Institute For the Application of Geospatial Technology*. Retrieved 01 26, 2012, from <http://www.iagt.org/downloads.aspx/>

Landsat Program. (2011, 11 27). Retrieved 01 25, 2012, from Wikipedia, the free Encyclopedia: <http://en.wikipedia.org/wiki/Landsat>

Liang, S. H., Chang, D., Badger, J., & Rezel, J. (2010). GeoCENS: Geospatial Cyberinfrastructure for Environmental Sensing. *Sixth international conference on Geographic Information Science*.

Na, A., & Priest, M. (2007). *Sensor Observation Service*. OpenGIS® Publicly Available Standard, Open Geospatial Consortium Inc.

NASA. (2012). *Earth Observatory*. Retrieved 01 26, 2012, from <http://earthobservatory.nasa.gov/Features/BlueMarble/>

NASA. (2009, 06 17). *Jet Propulsion Laboratory*. Retrieved 01 26, 2012, from <http://www2.jpl.nasa.gov/srtm/>

Nüst, D., Bache, F., Bröring, A., Stasch, C., & Jirka, S. (2010). Visualizing the Availability of Temporally Structured Sensor Data. *13th Agile Conference on Geographic Information Science*.

Olaya, V. (2007). SEXTANTE: a gvSIG-based platform for geographical analysis. *Proceedings of the Free and Open Source Software for Geospatial conference (FOSS4G)*. Victoria, Canada.

Percivall, G., Reed, C., & Davidson, J. (2007). *Sensor Web Enablement: Overview And High Level Architecture*. Open Geospatial Consortium Inc.

Robin, A. (2011). *OGC® SWE Common Data Model Encoding Standard*. OGC® Publicly Available Standard, Open Geospatial Consortium Inc.

Simonis, I., & Echterhoff, J. (2011). *OGC® Sensor Planning Service Implementation Standard*. Open Geospatial Consortium Inc.

Stephan Nebiker, M. C. (2006). INTEGRATING MOBILE GEO SENSORS INTO COLLABORATIVE VIRTUAL GLOBES – DESIGN AND IMPLEMENTATION ISSUES. *5th International Symposium on Mobile Mapping Technology*.

Tamayo, A., Viciano, P., Granell, C., & Huerta, J. (2011). Sensor Observation Service Client for Android Mobile Phones. *Workshop on Sensor Web Enablement 2011 (SWE 2011)*. Banff, Alberta, Canada.

The Globe Program. (2012). Retrieved 01 26, 2012, from <http://globe.gov>

The Kronos Group. (2012). *The Industry's Foundation for High Performance Graphics*. Retrieved 01 25, 2012, from <http://www.opengl.org/>

Tuttle, C., & D., E. (2008). How Virtual Globes are Revolutionizing Earth Observation Data Access. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* , 137-140.

Vretanos, P. (2010). *OpenGIS Filter Encoding 2.0 Encoding Standard*. OpenGIS® Implementation Standard, Open Geospatial Consortium Inc.

Vretanos, P. (2010). *OpenGIS Web Feature Service 2.0 Interface Standard*. OpenGIS® Implementation Standard, Open Geospatial Consortium Inc.

Wikipedia. (2011, 06 03). *Wikipedia, The Free Encyclopedia*. Retrieved 01 25, 2012, from Spot Image: http://en.wikipedia.org/wiki/Spot_Image

Wikipedia. (2012, 01 10). *Wikipedia, The Free Encyclopedia*. Retrieved 01 25, 2012, from http://en.wikipedia.org/wiki/Virtual_globe

WMO. (2012, 02 03). *World Meteorological Organization*. Retrieved 02 03, 2012, from Observations: http://www.wmo.int/pages/themes/observations/index_en.html

World Wind Central. (2009, 08 22). Retrieved from
<http://worldwindcentral.com/wiki/Add-on:ZoomIt!>