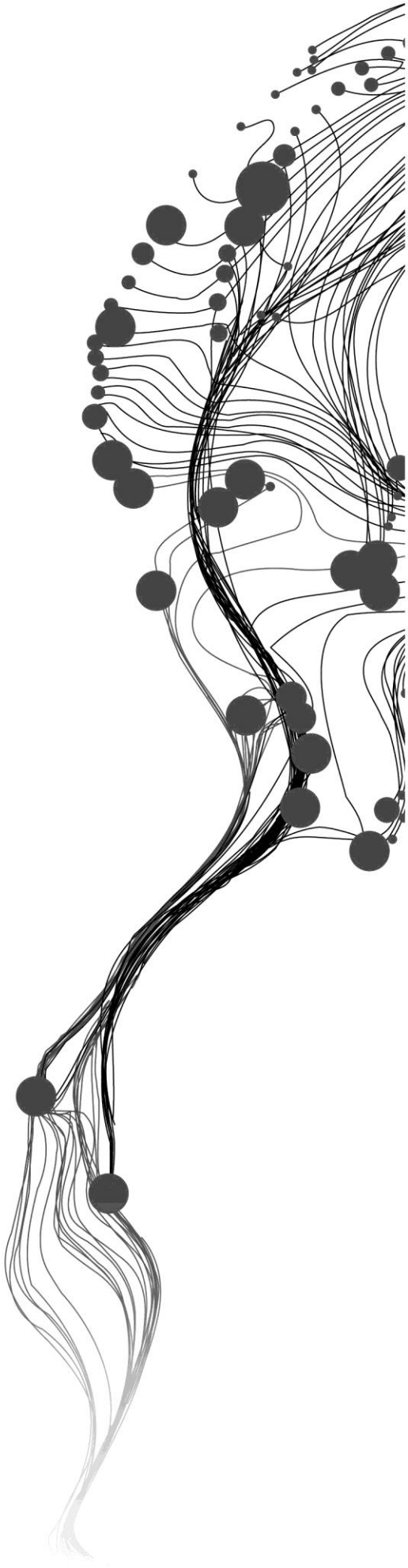
SERVICE-BASED SHARING AND GEOSTATISTICAL PROCESSING OF SENSOR DATA TO SUPPORT DECISION-MAKING

EDGARDO ALFREDO VÁSQUEZ GÓMEZ

March, 2015

SUPERVISORS:

dr.ir. R.L.G. Lemmens
dr. N.A.S. Hamm



SERVICE-BASED SHARING AND GEOSTATISTICAL PROCESSING OF SENSOR DATA TO SUPPORT DECISION-MAKING

EDGARDO ALFREDO VÁSQUEZ GÓMEZ
Enschede, The Netherlands, March, 2015.

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
dr.ir. R.L.G. Lemmens
dr. N.A.S. Hamm

THESIS ASSESSMENT BOARD:
prof.dr. M.J. Kraak (chair)
dr. S. Jirka, 52°North (external examiner)

ABSTRACT

Sensor networks are used frequently to monitor environmental variables such as air pollution in an increasing number of cities around the world. However, the monitoring stations are often limited in quantity and spatial resolution, thus air pollution concentrations need to be predicted for the locations where it is not been measured. In this regard, as geostatistics is an area of statistics through which values at unsampled locations may be predicted using the known measurements taken at the monitored locations, it may be used to tackle the aforementioned sensor network's coverage. On the other hand, the data retrieved from the monitoring stations as well as the predicted values are required to be available for the users in nearly real-time. Hence, it is also necessary to automate the sensor data transportation, the geostatistical processing and in some cases the data pre-processing stage. Sensor data may be provided in a wide variety of formats, depending on the monitoring equipment vendors or the nature of the measured phenomena. Such data heterogeneity might become a problem of interoperability among the sensor network and the people interested in getting the retrieved measurements. Hence, it is necessary to have a suitable means for sharing both the data from the sensor network and the predicted values for the unsampled locations. This study considers the Open Geospatial Consortium (OGC) initiative for standardizing the communication rules and the sensor data formats, as the basis for the design and implementation of an interoperable platform for sharing and performing geostatistical functions on sensor data through web services. A prototype implementation has been realized in order to determine the feasibility of developing such platform and air pollution (PM10) data have been used as input to automatically perform spatial predictions at the unsampled locations. Additionally, the predictions' quality has been assessed so it can be determined whether they can be used to support decision-making processes.

Keywords: Sensor data; Geostatistics, Automatic spatial interpolation, Web service, Air pollution.

ACKNOWLEDGEMENTS

I would like to express my most profound gratitude to God, my family and my friends for their constant and unconditional support and encouragement.

I would also like to thank all the people that have made the achievement of this academic goal possible and have helped me along this experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and problem statement

In recent times, several organizations are making efforts to implement the so called 'ubiquitous computing' (Weiser, 1993) among the urban elements such as people, infrastructure and open spaces by using sensor networks (Kumar, 2003) to capture data continuously and to share these data as described in (Ho Lee, et al, 2008).

Nonetheless, this fact has certain implications and limitations to be considered, as mentioned in (Huang & Tseng, 2005): the high investment required not only to acquire but also to keep a sensor network running and capturing data in such a way that they have the desired quality; the high and continuous energy supply required; and the risk of having the sensors exposed to different atmospheric conditions, among others.
In this regard, the increasing utilization of mobile devices across a city is an opportunity for delivering information to citizens as well as collecting data from them. Although, mobile devices' presence in most of the cities is not yet high enough to consider it as the seamless computer-enabled environment described by (Poslad, 2009) as 'ubiquitous computing'.

This is why, for the present study any dataset collected either through sensors or mobile devices can only be considered a limited sample and therefore it is considered necessary to apply geostatistical functions on this sample in order to derive information about the whole study area; such processes as well as the means through which the data can be distributed and accessed were implemented on web services.
The study also considers three groups of users: 'common users', the people needing sensor data when planning outdoor activities but with no high knowledge of geospatial analysis; 'advanced users', people needing spatial prediction models to support decision making processes; and 'developers', the people in charge of adding components to the platform or replacing the existing ones. Further description of each group's members can be found in section 3.3 (User types).

Moreover, a case study was held in the city of Eindhoven, The Netherlands, and on the data provided by this city's 'AiREAS' project; which is offering access to air quality measurements collected through monitor stations called 'Airboxes' (see Section 3.1). Besides, a service-based platform was designed and tested to provide nearly real time values to the types of users mentioned above.

The limited number of in-situ sensors available leads to one of the main problems regarding a sensor network (Huang & Tseng, 2005): its coverage in some areas of the city. Thus, geostatistical processing methods are needed before delivering the sensor data, in order to derive the predicted values to produce the air pollution map as well as its corresponding uncertainty map. Besides, automating these sharing and geostatistical processes is necessary, as air quality is variable in space and time, to deliver nearly real-time values.

Eindhoven 'AiREAS' (AiREAS, 2014) as well as similar organizations interested in delivering information to users and collecting data from them, plus processing these data and sharing the final outcome, have to deal with different technologies and methods for each stage. Such methods are not necessarily aiming to work together, and this fact leads to interoperability issues like the inability to provide or accept services on one of the components (sensor network, server, client devices).

Thus, it is necessary to design a standards-based 'integration platform' which can facilitate the whole workflow described above, in a way that it can be perceived as one single solution and it can be used for all these organizations to accomplish their sharing and processing goals; e.g. retrieve the data from sensors to a server, then the processed data from a server to client interfaces. In order to design such platform, it was necessary to analyze the different available methods, tools and data formats on the pursuit of interoperability, efficient data sharing and automatic geostatistical processing. Thus both the coverage problem and the interoperability issue can be tackled.

1.2. Research identification

As a starting point to support the above mentioned platform, the present study proposes the implementation of web services to retrieve the raw data from the sensor network, pre-process the data in such a way that they can be standards compliant, and perform automatic geostatistical processes and share the outcomes, using standards-based technologies in order to ensure interoperability among the different platforms for: server, clients and the sensor network.

It is also required to determine whether it is possible to build a reliable spatial prediction model from the sensor network data, performed by deriving air quality values, for areas with no coverage or facing problems, whether it is possible to test this model's quality and if the outcomes can be delivered through client applications.

In summary, there are three considerations included in this study and they can be described as follows:

- Web technologies and standards for allowing interoperability among sensor networks, server and the client side and for sharing the data through the different platforms involved.
- Geostatistical functions, such as spatial interpolation and cross-validation, used to produce the model to derive values for the areas with coverage problems and for assessing the model's reliability.
- Web services and tools required for automating the required spatial interpolation processes and for sharing the outcome.

1.3. Research objectives

The core aim of this work is to design and implement a platform to get air quality data from an in-situ sensor network, automatically perform spatial predictions from these data and share the resulting air pollution map and its corresponding uncertainty map through client devices in nearly real-time. The following objectives and questions guided this research:

1. To design and implement an interoperable and standards-based platform for sharing and geostatistical processing air quality data.
2. To determine how spatiotemporal functions in general and spatial prediction in particular can be used to perform geostatistical processing on air quality data retrieved from sensor networks.
3. To provide a set of standardized functions that can be executed on web services to automatically perform spatial prediction and to share the outcome with the client side.

1.4. Research questions

Related to objective 1:

1.1 What are the most suitable data models to share and process air quality data through web services?
1.2 Which architecture is appropriate to share properly the performed spatial predictions?
1.3 What are the rules to ensure interoperability among the different platforms involved in this process?

Related to objective 2:

2.1 Which functions are most suitable to perform spatial prediction on air quality data retrieved from sensor networks?
2.2 How can the uncertainty of the spatial prediction results be determined?
2.3 Which functions are the most appropriate to assess the outcome's quality?

Related to objective 3:

3.1 Can web services be effectively used to receive data from a sensor network and to share the outcome through client applications?
3.2 How should the outcome be communicated to the different user types?
3.3 What is the extent to which this process can be automated?

1.5. Innovation aimed at

This research has the aim of providing an integrated platform to support decision-making by effectively combining methods for: processing and sharing air quality data, performing automated spatial prediction models on retrieved data and facilitating the communication among a sensor network, server and client sides through standards-based web technologies.

In addition to the above described combination of methods, innovation lies in the following characteristics:

- Heterogeneity of the data domains (such as sensor data, spatiotemporal data, etc.), leads to the necessity for different analyses to be realized on each type of data utilized.
- In order to support decision-making, the data delivered to the users must be available in nearly real-time, thus involving automatic web-based processing by using atomic functions and temporary structures for data storing, such as buffers.
- Data sharing and processing must be standards compliant, so these data as well as the results can be used for different processes and analyses.

1.6. Thesis outline

This thesis consists of seven chapters. **Chapter 1** explains the motivation and problem statement, and describes the research objectives and questions. **Chapter 2** gives an overview of some related works and the characteristics and special requirements for designing a service-based platform. **Chapter 3** describes the case study and the dataset to be utilized for the present work. **Chapter 4** presents the proposed approach to design a platform for sharing and performing geostatistical functions on sensor data. **Chapter 5** explains the prototype development based on the adopted approach as well as the obtained results. **Chapter 6** contains a discussion on the prototype implementation results. Finally, **Chapter 7** presents the conclusions of this study and some recommendations for further research.

2. SERVICE-BASED PLATFORM COMPONENTS CHARACTERIZATION AND REQUIREMENTS DEFINITION

Before starting the design of a platform capable of handling sensor data, it is important to consider the peculiarities of these data in order to define the basic requirements for the proposed platform; it is also important to consider some previous efforts in this regard. This chapter gives a summary of these efforts and goes through a revision of the characteristics that have to be taken into account when dealing with both sharing and processing data of such type.

2.1. Related work

Several spatial data integration issues are highlighted in (Mohammadi, et al, 2010) which also includes a tool for evaluating technical and non-technical characteristics of spatial datasets; however, there is no proposed solution for the discussed problems such as bottlenecks and dataset inconsistencies. The work described in (Juba, et al. 2007) includes an integration of software packages and web technologies for interactive mapping, but neither coverage issues of sensor networks nor mobile platforms are included. The general aim of this study has two relevant aspects that are considered separately as follows:

2.1.1. Data sharing

The need for sharing and visualising the data can be tackled by using the available geospatial web services (Granell, et al, 2007). In like manner, the currently available client platforms make it possible to configure an application which can play two roles: to deliver and to capture data as required in this study.

Due to the fact that sensor data is one of this research's data sources, the approach proposed and implemented by (Foerster, et al, 2012) is also relevant. Its aim is to discover data and services in the so called Sensor Web through mobile applications and to describe the use of Sensor Web Enablement (SWE) as the Open Geospatial Consortium (OGC) initiative for standardising access and publishing of sensor data. It is also a revision and classification of a number of approaches about ubiquity as one of the goals for developing a context-awareness systems (Strang & Linnhoff-Popien, 2004).

It is also relevant to mention the service prototypes described by (Havlik, et al, 2009), which extend the usability of the OGC SWE architecture through the development of special services such as the so called "Cascading SOS" (SOS-X): a client to the underlying OGC Sensor Observation Service (SOS) that provides alternative access means to users or services, plus the capability of re-formatting, re-organizing and merging data from several sources into a single SOS.

2.1.2. Geostatistics and spatial modelling

Geostatistics contributes to describe variables distributed either in a spatial or in a spatiotemporal domain (Chilès & Delfiner, 2009); it can be very useful to perform spatial interpolation to predict the air quality values at unsampled locations. This study can be carried out by considering geostatistical processing as air quality data is spatiotemporally distributed.

An alternative approach is the presented by (da Cruz, et al., 2013) who introduce the concept of 'quality maps' and their contributions to rank stochastic realizations as well as incorporating uncertainty into decision making, which is an important part in this study, since the outcome's quality is relevant to support decision-making processes.

A number of tools have been developed to support geostatistical process and analysis, for instance: 'INTAMAP' (Pebesma et al., 2011), which can be used for automatic mapping and consumed as a web service, as well as giving the possibility to assess models' quality.

Additional interpolation performance assessment, based on 'INTAMAP' Web Processing Service is presented in (de Jesus, et al, 2009) concluding with a discussion on the use of k-fold cross validation and discussing on its limitations compared with the 'INTAMAP' interpolation service. This fact should be considered during the present case in order to enhance the process efficiency.

2.2. Characteristics of sensor data

Sensor data have a number of special characteristics that must be considered in order to design a suitable integration platform which allows handling them efficiently.

Sensor data are strongly correlated in space, as they describe heterogeneous spatiotemporal physical phenomena (Jindal & Psounis, 2004); and they have an important and massive growth rate due to the quantity and quality of data elements, as they might include not only scalar values but also multimedia content (Akyildiz, et al, 2007).

Besides, sensor data might be expressed in different spatial or temporal resolution, e.g. the geographic extent, the number of nodes or the sampling frequency (Ganesan, et al, 2003); and they might be distributed around a sensor network formed of a number of monitoring stations interconnected with different platforms for storing and processing them (Aberer, et al, 2007).

2.3. Characteristics of data models

It is relevant for this study to consider different data models used for storing and transporting vast amounts of data, since sensor data tends to increase indefinably, and eventually to reach vast levels. In this regard, most commonly two main paradigms are being used for this purpose: Relational databases and Non-Relational databases, the so-called NoSQL or Not Only SQL.

2.3.1. Relational model

The relational database (RDMS) approach has been the dominant model during the last decades (Nance, et al., 2013), as it provides a number of services and tools addressing a wide variety of requirements and supporting the most important business tasks.

As discussed in (Atzeni, et al., 2014) some RDMS features to take into account are: transaction processing, analytical support and decision support tools; SQL is also a standard language (even though it has some dialects that differ among vendors) allowing it to provide reasonably general-purpose solutions.

Moreover, as it is the most proven approach to store and query data (Lee, et al., 2013) it can be established that it has been used effectively for many traditional enforcements and it can deal with complex

transactions and queries which are supported by an extensive set of tools that lead to a robust solutions implementation and maintenance.

Additionally, the Atomicity-Consistency-Isolation-Durability (ACID) attributes are guaranteed by the normalization process as it is necessary to ensure that when a transaction is finished the database remains in a consistent state (Vogels, 2009), thus helping to ensure the data reliability.

Nevertheless, there are also certain drawbacks related to the use of RDBMS as the data storage model. As discussed in (Lee et al., 2013) it is not completely practical for certain forms of data requiring a large number of fields to handle different types of data involved when very often these are partially unused, leading to an inefficient storage and a poor performance.

It also makes completely necessary to pre-design the exact field structures of data, which despite of being effective for many traditional enforcements, is considered to be too rigid or not useful for some other cases, e.g. when it is necessary to model a dynamic entity having certain attributes that are only required during a certain period of the year, thus all the attributes need to be created despite the fact that they are not used most of the time.

Furthermore, neither the transactions nor the queries are as complex as assumed for certain contexts, so they do not need to be supported. And it uses large monolithic architectures instead of the scale-out models used nowadays in the pursuit of flexibility.

2.3.2. Non-Relational Databases

The non-relational model consists of a set of data manipulation techniques and processes which do not use the table-key model (i.e. the one used by RDMS). The so-called NoSQL (Not Only SQL) is the most popular database model based on this; it is a distributed database system which does not require fixed tables schemas, does not use join operations, among other distinctions discussed by (Tudorica & Bucur, 2011) and classified as follows.

Core NoSQL systems

Most of them are created as component systems for Web 2.0 services. The following subtypes are recognized in (Moniruzzaman & Hossain, 2013):

Wide column storage: they use a distributed and column-oriented data structure in which, every item is stored as a pair formed of an attribute name and its value; each record can have a different number of columns and its columns can be nested (super columns can be created) as well as grouped (column families) in order to access them. Stored data can be retrieved by primary key, per column family.

Document storage: they were designed to manage and store documents, typically using a JSON-like structure (i.e. encoded in a standard data exchange format like XML, BSON or JSON). Since each document is in fact an object, it is closely aligned with object-oriented programming. The value column contains semi-structured data (attribute name – value pairs) and the documents contain at least one field of certain typed value (string, date, binary, array, etc.); each record and its associated data are stored in a single document and both keys and values can be used for searching.

Key value storage: they allow the retrieval and updating of data based only on a primary key. Besides, they offer very limited query functionality and they might imply an extra development cost and application level requirements, for instance: two round trips might be required to perform an update, the first one to find the record and the second to update it.

Graph databases: they replace relational databases with graph structures with nodes, edges and properties to represent data with interconnected key-value pairings. Then, data is modelled as a network of specific elements and their relationships; it can be used for an extended number of applications but its comprehension is time consuming.

Soft NoSQL systems

They are most commonly not related to any Web 2.0 service but they have NoSQL features. Even though, some of them have relational capabilities, like Atomicity, Consistency, Isolation and Durability (ACID properties). That is why they are often excluded of the list of NoSQL systems. There exist the following subtypes:

Object databases: the study carried by (Schmidt, et al, 1988) establishes that the main reasons of the acceptance of this model are the conceptual naturalness of it, as well as the programming languages and software engineering trends. Besides, the correspondence study between Object and Relational Models reveals the similarities and relatively short transition process needed regarding some aspects of conceptualization and realization of a model and its migration from one to the other; although not all of these aspects are that suitable, since the data manipulation depends on SQL this transition requires a rather complex process.

Grid & Cloud databases: they provide data structures that can be shared among nodes in a cluster and distribute workloads among the nodes (Perumal Murugan, 2013), providing also a framework for securing read or write client operations. There are data grid platforms based on this model that support shared data structure, highly concurrent and cache capabilities; maintaining state information among nodes through a peer-to-peer architecture and also supporting data storing in the cloud (either private or public).

XML databases: also known as document-centric, they are databases that support eXtensible Markup Language (XML) format for storing or use XML documents as input or output. Typically, when the data are stored in XML format they use XPath and XQuery to support the queries. Some of them guarantee ACID properties for data store have client-server architecture, are fully indexed and highly scalable.

2.3.3. Hierarchical data models

This is based on a conceptual model which establishes that data are organized into records that are recursively composed of other records (Liu & Özsu, 2008) all connected by links; data is organized in a tree-like structure. A record is a collection of files, each of them containing one value, this value corresponds to a tuple (row) in the relational database model and the value plus a relation (table) is equivalent to the so called entity type, the definition of which field a record contains.

Two implementations of this model, that are being used to handle very large sensor datasets, are: Hierarchical Data Format (HDF) (HDF-Group, 2014); and Network Common Data Form (NetCDF) format (UCAR, 2015); both having platform-independent libraries that support the creation, access and sharing of data.

2.4. Characteristics of geostatistical models and functions

Geostatistics are used, among other activities: to explore and describe spatial variation in sensor data (Curran & Atkinson, 1998), to increase the accuracy with which sensor data can be used to estimate continuous variables, and to model the uncertainty about unknown values (Goovaerts, 1997).

Moreover, geostatistics helps to overcome the need for making predictions of sampled attributes at unsampled locations from sparse data, often implying high cost acquiring processes (Burrough, 2001); in particular it provides reliable interpolation methods with uncertainty assessment means; useful methods for generalization, upscaling and for supplying multiple realizations of spatial patterns that can be used in environmental modelling.

2.4.1. Geostatistical models

Geostatistical space-time models are used to address dynamic processes evolving in space and time (Kyriakidis & Journel, 1999) such as environmental sciences, global warming, hydrology, etc. and their need to make predictions of sampled attributes at unsampled locations (Burrough, 2001). In this regard, there are two main types of models to be considered for this study:

Deterministic models

These models have no probabilistic elements and the models' input and output relation is conclusively determined, i.e. selected uniformly and independently over a given area; this fact means that they do not take full advantage of the spatial information available (Rossi, et al, 1994). Often, these models also require a large amount of input parameters which are not easy to obtain because sensor data involve rather limited or indirect sampling (Kyriakidis & Journel, 1999), mainly in cases involving environmental measurements.

Stochastic models

They model spatiotemporal behaviour of phenomena with random components; because, in several cases it is difficult to build an intuitive perspective, they aim at building a process that only imitates some patterns of the observed spatiotemporal variability (Kyriakidis & Journel, 1999). Hence, when a model includes the concept of randomness and provides both: estimations (deterministic part) and associated errors (stochastic part) assessment, i.e. when uncertainties are represented as estimated variances, such model is stochastic; otherwise it is deterministic.

Therefore, stochastic models are more general than geostatistical models, though they both describe stochastic phenomena. However, stochastic models emphasize the modelling process, whereas geostatistical models' emphasis is on data analyses (Coburn, et al, 2005). For this study, stochastic models are used with only one variable (location: u) as this project only covers the spatial domain of the phenomena described by the considered (air quality) sensor data.

Thus, the general model of a spatial process Z(u) (Equation 2.1) shows the mean dependence on location $\mu(u)$ the spatially correlated error S(u) and the spatially uncorrelated error e.

$$Z(u) = \mu(u) + S(u) + e \quad (2.1)$$

However, there is an important assumption in this method: the unknown spatial mean (to be estimated) over the study area is constant; this constant mean assumption, represented by equation (2.2) is considered in equation (2.3).

$$E[Z(u)] = \mu \quad (2.2)$$

This fact leads from the general model shown in equation (2.1) to the one in equation (2.3); where μ is the location independent trend, S(u) is the spatially correlated error and e is the spatially uncorrelated error.

$$Z(u) = \mu + S(u) + e \quad (2.3)$$

2.4.2. Interpolation

The air quality data considered in this study, form a regionalized variable, they are consistent with the definition given by (Chilès & Delfiner, 2009) i.e. there is a numerical function, depending on a continuous space index and combining high irregularity of detail with spatial correlation. And, it is necessary to estimate air quality values at places where it is not been measured, these places can be seen as nodes of a regular grid. Thus, it can be used the process known as "gridding" or interpolation.

The different techniques utilized to perform spatial interpolation may be categorized in several ways (Li & Heap, 2008 and Franke, 1982), including: gradual or abrupt, the produced surface might be smooth or discrete, it depends on the criteria (simple distance relations, minimization of variance, etc.) used in the selection of weight values in relation to distance; and univariate or multivariate, the methods that only use samples of the primary variable for deriving estimations are univariate and the ones that use secondary variables are multivariate.

Kriging is a geostatistical interpolation technique which has the aim of getting estimations that are not systematically too high or too low (unbiased), as well as quantifying the precision of the estimations by obtaining the error variance or its square root the standard error (Chilès & Delfiner, 2009).

The different kriging methods depend on the underlying model, determined during the structural analysis phase, through the variogram as the expression of the spatial variability. These methods have been used commonly in environmental assessment, among other fields (Bayraktar & Turalioglu, 2005), because they give the possibility of determining the predictions uncertainty.

2.4.3. Variogram

The variogram function describes the spatial dependence of a spatial random field or stochastic process Z realized at two locations (u) and (u+h); this function can also be defined as the variance of the difference between field values at two locations across realizations of the field (Cressie & Cassie, 1993). The variogram's parameters are as defined by (Cressie, 1988):

- Nugget, the value at which the variogram intercepts the y-axis. If it is not zero there is a nugget effect, formed of the covariance due the micro-scale variations (C_{MS}) and the covariance due to the measurement error (C_{ME}).
- Sill, the variogram's upper limit composed of a partial sill estimate, a nugget effect estimate and the range.
- Range, the distance at which the semi-variogram reaches the sill and the smallest lag (h) at which the measured observations Z(u) and Z(u+h) are correlated.

Stochastic models apply spatial correlation represented by the variogram, also referred as semi-variogram by several authors (Bachmaier & Backes, 2008), relating uncertainty with the distance between observations. In order to represent such spatial correlation, the variogram is obtained from the data by using the so called semi-variance (Equation 2.4), a function of the separation distances (h) between observations (i.e. the lag).

$$\Upsilon(h) = \frac{1}{2} E\,[Z(u) - Z(u+h)]^2 \qquad (2.4)$$

The semi-variance can be described as the expected squared difference between pairs of points separated by certain distance, divided by two (Chilès & Delfiner, 2009).

2.4.4. Accuracy assessment

Cross-validation is a model validation function, used to estimate the accuracy of a prediction model. The diagnostic measures can be computed from cross-validation's result through: Root Mean Square Error (RMSE), Mean Error (ME) and Mean Square Deviation Ratio (MSDR) of the residuals.
Let $Z^*(u_i)$ be the predicted value and $Z(u_i)$ the observed (known) value at location u_i; and N the number of values in the dataset:

$$\mathrm{ME} = \frac{1}{N}\sum_{i=1}^{N} Z^*(u_i) - Z(u_i) \tag{2.5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Z^*(u_i) - Z(u_i))^2} \tag{2.6}$$

$$\mathrm{MSDR} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(Z^*(u_i)-Z(u_i))^2}{\widehat{\sigma^2(u_i)}}} \tag{2.7}$$

Ideally, Equation 2.5 should give 0, because kriging is the best linear unbiased predictor; although, it is a weak diagnostic measure for kriging as it is insensitive to inaccuracies in the variogram (Robinson & Metternicht, 2006). And, Equation 2.7 should give 1 as the result, because the cross-validation residuals should be equal to the prediction errors at each point; if it is greater than 1, the predictions' variability is underestimated and vice versa (Robinson & Metternicht, 2006).

The ordinary kriging variance: $\sigma^2(u_i)$ is also part of Equation 2.7, it may be obtained as indicated by Equation 2.8. The first term in this equation is the covariance with a lag equal to 0; the second one reduces the prediction uncertainty by using correlation with neighbour points; and the third one increases the uncertainty because of the mean estimation uncertainty.

$$\widehat{\sigma^2}(\hat{T} - T) = C(0) - c_0^T C^{-1} c_0 + \left(1 - c_0^T C^{-1}\right)^T (1^T C^{-1} 1)^{-1} (1 - c_0^T C^{-1} 1) \tag{2.8}$$

2.5. Characteristics of service-based platforms

A 'Sensor Network' is formed of a number of spatially distributed sensor resources that have communication among them, measuring and relying information about the phenomenon to the observer (Tilak, et al, 2002). A 'Sensor Web' is an infrastructure that enables interoperability among sensor resources (discovering, accessing, tasking, eventing and alerting); hiding the underlying layers which allow communication among heterogeneous hardware and different sensor networks (Nittel, et al, 2008), from the application level.

Sensor Web Enablement (SWE) is one of the Open Geospatial Consortium (OGC) initiatives for establishing the interfaces and protocols to implement a 'Sensor Web' through which applications and services are allowed to access any type of sensors over the Web (Řezník, 2007). Thus, SWE specifications provide the functionality to integrate sensors into Spatial Data Infrastructures (SDI) in the standardized way described in (Granell, et al, 2009) to couple sensor data with spatiotemporal resources at the application level.

In order to manage the heterogeneity of the aforementioned sensor resources, it is necessary to use certain technologies as the middleware, just like the Sensor Web, between these resources and the applications. According to (Bröring et al., 2011) these technologies can be classified as follows:

2.5.1. SWE service specifications implementations

The Open Geospatial Consortium (OGC) Sensor has developed the Web Enablement (SWE) initiative as a suite of standardized web-service interfaces and XML schemata that allow live integration of heterogeneous sensor webs into an information infrastructure (OGC, 2015c).
Sensor data, as established in Section 2.2 (Characteristics of sensor data) are most likely heterogeneous and profuse, thus sensor data integration is a laborious task often including certain data conversions and transformations that might imply information loses (Havlik, et al, 2009). This situation has led to the establishment of regional, continental or global directives and initiatives aimed to facilitate a seamless information exchange. Some examples of these initiatives are:

"Infrastructure for Spatial Information in the European Community" (INSPIRE) directive which demands the use of spatial information services to exchange geo-referenced environmental information (INSPIRE-Directive, 2007).

"Global Monitoring for Environment and Security" (GMES), the European contribution to the Group on Earth Observation (GEO) and its implementation plan for an integrated Global Earth Observation System of Systems (GEOSS) (Scholes et al., 2008).

In addition to this regulatory documentation, the European Union has sponsored the ORCHESTRA Integration Project, an information infrastructure implementation and research project that has defined the "Reference Model for Orchestra Architecture" (RM-OA) (Usländer, 2005) as well as the associated services and specifications for their implementation on different technology platforms. Besides, the "Sensors Anywhere" Integration Project (SANY IP) has extended RM-OA by including sensor and sensor network specific services and processing.

Another important extension of RM-OA in the area of in-situ monitoring, is given by the Sensor Service Architecture (SensorSA) (Usländer et al., 2009), which is a Service-Oriented Architecture (SOA) that includes elements of Event-Driven Architecture (EDA); and a particular focus on the access, management and processing of sensor data.

This extension is achieved through the inclusion of the specifications defined by the SWE (Botts, et al., 2008); as well as the definition of the data models and interaction patterns required for such in-situ monitoring. Thus it has been used as basis for interoperability feasibility projects such as SensorSA, which embraces the OGC SWE framework of open standards; specifically the Sensor Observation Service (SOS) is used to publish observations from sensors and other sensor-liked data sources (Havlik et al., 2009).

There are solutions designed for making sensors available on the web as well as for allowing the access to them from the application level through sensor web infrastructures, based on the SWE specification. They typically do not provide managing functionalities because they use SWE standards to allow the interoperable access to the sensors. Some examples are:

52º North sensor web framework: it provides implementations for the different SWE services, as Sensor Observation Service (SOS) which enables querying and inserting measured sensor data and metadata; Sensor Event Service (SES) which pushes sensor data in case of user defined filter criteria; Sensor Bus which integrates the sensor resources with the SWE service implementations in such a way that they are adapted to each other and have communications among them.

GeoSWIFT: the main difference with the one mentioned above is, its peer-to-peer based spatial query framework, introduced to optimize its scalability.

PulseNet: it is a modification of the open source 52° North Sensor Web Framework components that allows accommodating legacy and proprietary sensors in SWE-based architectures.
NASA's sensor web: it incorporates SWE services and combines them with the Web 2.0 technology to allow the creation of mash-up applications to integrate data from multiple sources.

2.5.2. Non-standardized approaches

There are other solutions designed with the same goal of allowing the access to sensors from the application level. Nonetheless, they do not use SWE standards and specifications but instead they define their proprietary interfaces and data encodings. Besides, they do not offer service interfaces for sensor tasking. The following are examples of these:

Global Sensor Network (GSN): its main focus is on a flexible integration of sensor networks for enabling fast deployment of new resources; the core concept behind it is the virtual sensors (e.g. simulations) abstractions with XML-based deployment descriptors in combinations with data access through plain SQL queries.

Hourglass: It provides the architecture for connecting sensors to applications and offers the sensor discovering and data processing services while maintaining the quality of them at the same level as is presented on data streams.

Sensor Network Services Platform (SNSP): It defines a set of service interfaces usable as an Application Programming Interface (API) for a Sensor Network independently of particular implementations or hardware platforms. It also has non-standardized service interfaces for data querying and sensor tasking, and also auxiliary location and timing services, as well as a concept repository.

SOCRADES: It comprises multiple services (like discovery, eventing and data access) and it also provides sensors integration into an infrastructure through the implementation of sensor gateways. Though, as expected the individual services operations are not standardized.

2.6. Service-based platform requirements

All the above mentioned characteristics are considered in this study to address the research questions related to the system's architecture and the involved data models, such characteristics lead to special requirements for dynamic integration and sensor data handling through a service-based platform which are part of the proposed system's design and the prototype's implementation.

1. Regarding their spatiotemporal nature, it is necessary to ensure that the long term storage means as well as the communication technology to be used guarantees the capability of performing spatiotemporal queries needed for pattern mining (Ganesan, Estrin, et al., 2003).
2. Since, it is necessary to determine the existence of spatiotemporal correlations among the sensor data, the platform must provide such analysis tools (Ganesan, et al, 2003).
3. Sensor data storage must operate with resource utilization efficiency and optimization, providing also compression capabilities in order to deal with the big amount of data as well as the high rate of message transmission (Wang, et al, 2008).

4. The heterogeneity and multi-resolution nature of sensor data implies the necessity of having access to the hierarchical and multi-dimensional structures designed to store them (Diao, et al, 2007).
5. The platform design must consider the need for distributed work-load processing in order to support the data handling and transportation throughout a sensor network (Aberer et al., 2007) regardless of its extent.
6. Sensor data might be required either synchronously or asynchronously (Alouani & Rice, 1998) as there are different sensor data rates with inherent communication delays between sensor platforms and remote processing sites.

3. CASE STUDY AND DATASET DESCRIPTION

Since recent years, the city of Eindhoven has established the social goal of becoming a city in which anyone can enjoy the clean air while practicing sports or any other outdoor activities with no health problems caused by air pollution. That is why citizens, businesses, academic institutions and local government are assuming responsibilities and working together as active part of the AiREAS project (AiREAS, 2014).

The project is being carried out through the so called Intelligent Measurement System (ILM), carrying out health related research on air quality measurements. ILM consists of 41 air pollution monitoring stations, so called 'Airboxes' in different locations throughout the city, equipped with sensors that allow them to measure various types of particulate matter presence in the air (like PM1, PM2.5, and PM10) as well as ozone, relative humidity and other air pollution indicators.

The ideal scenario includes sharing the sensor data in real time, so the citizens and the enterprises can make quick decisions about their current activities; and the researchers involved in the quality of measurements can take immediate actions in case of failure on the sensor network.

Nevertheless, designing a completely functional communication platform is required in order to ensure the proper and continuous data sharing even when failures on the sensor network occur. Besides, measurements are now represented by points where an 'Airbox' is located and citizens require values for the regions around each 'Airbox', so they can plan a route for their outdoor activity; thus interpolation is needed over the set of measurement points to derive values for the rest of the city.

Moreover, the raw data produced by the 'Airboxes' need the automation of certain pre-processing so that potential gaps in the measurement routines, caused by technical failures, can be fixed; the wrong measurements produced during the failure can be filtered; and the data formats can be standardized.

3.1. Study area

The city of Eindhoven, in The Netherlands, comprises the study area. It is located at 51°26'N and 5°29'E. The location of the 41 'Airboxes' is shown in figure 3.1. Eindhoven's 'AiREAS' project (AiREAS, 2014) and its sensor network formed of 41 'Airboxes' with air quality information, are the main data source.

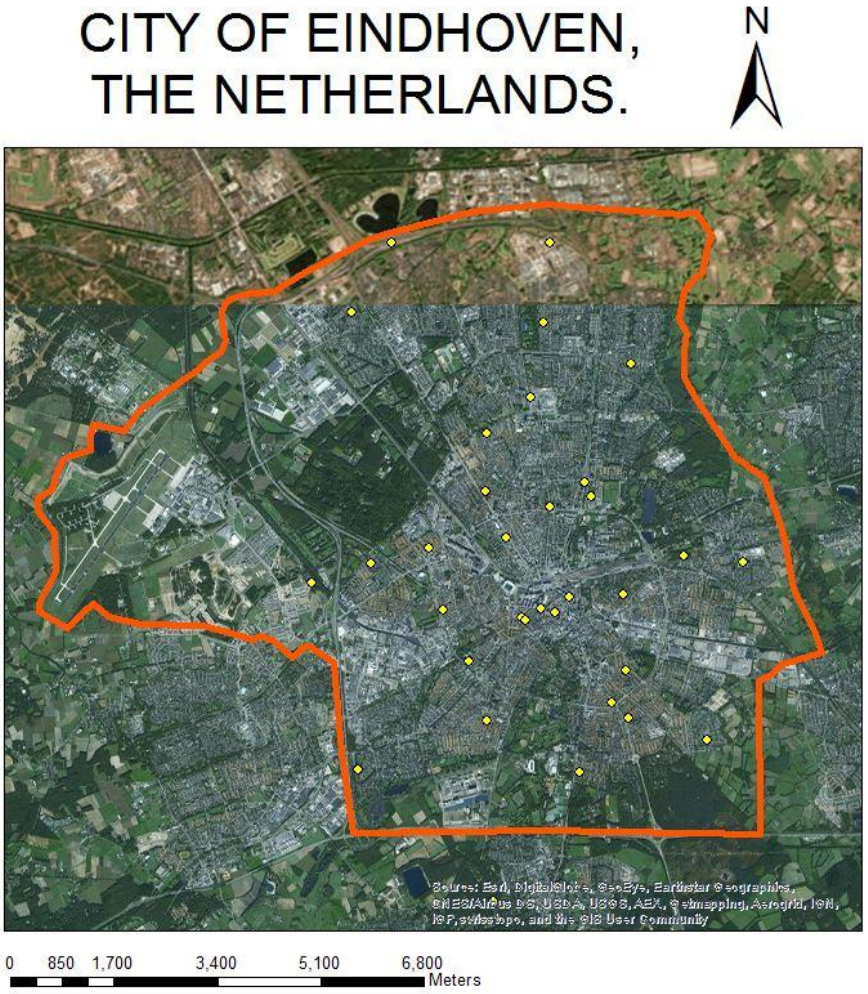


Figure 3.1 Eindhoven city boundary (orange) and 'Airboxes' locations (yellow)

3.2. Data description

The sensor data consists of records updated hourly with measurements for the elements listed in table 3.1.

Element	Description	Units
PM_1	Particulate Matter, up to 1 micron in diameter	$\mu g/m^3$
$PM_{2.5}$	Particulate Matter, up to 2.5 microns in diameter	$\mu g/m^3$
PM_{10}	Particulate Matter, up to 10 microns in diameter	$\mu g/m^3$
UFP	Ultra-fine particles, up to 0.1 microns in diameter	$\mu g/m^3$
Ozone	Ozone microns	$\mu g/m^3$
North	GPS geographic coordinate	DMS
East	GPS geographic coordinate	DMS

Table 3.1 Airboxes data catalog

The data gathered from the sensor network is stored daily in a folder, whose name indicates the corresponding date as well as the sensor ID; they are available through the project's public server in order with the IP: http://193.172.204.137:8080.
Every night, the last day's records are added to the measurement data, composed of a list of folders, the so-called batches, which are named like the following examples:

2013-11-02T000103 /
2013-11-03T000104 /

The batch with the current date is empty until one minute after midnight, when it is completed; then for each directory, a file called 'acquisition.h5', a Hierarchical Data Format version 5 (HDF5) file, containing all the calibrated values collected from all the 'AirBoxes' is available for download.
These records are also available as CSV files (one per 'Airbox') on: http://193.172.204.137:8080/csv/, each CSV file has the structured described in Figure 3.2 and includes all the fields' values as numbers, except for the 'time' field which is date-time.

EAST	NORTH	NOTUSED	OZONugperm3	PM10ugperm3	PM1ugperm3	PM2.5ugperm3	RELHUMpct	TEMP C	time
530.825596	5125.083657	0	0	0	0	0	0	0	01/01/2000 0:00
530.82311	5125.080491	0	0	0	0	0	0	0	01/01/2000 0:00
530.831284	5125.080254	0	0	0	0	0	0	0	01/01/2000 0:00
530.823136	5125.073471	0	0	0	0	0	0	0	01/01/2000 0:00
530.822746	5125.076488	0	0	0	0	0	0	0	01/01/2000 0:00
530.823105	5125.083682	0	0	0	0	0	0	0	01/01/2000 0:00
530.823105	5125.083682	0	0	0	0	0	0	0	01/01/2000 0:00
530.819719	5125.08328	0	0	0	0	0	0	0	01/01/2000 0:00
530.827685	5125.078658	0	323.64	11.66	2.27	4.31	112.95	10.67	01/11/2013 9:40
530.829921	5125.079646	0	335.57	9.15	2.45	3.64	112.29	10.79	01/11/2013 9:50
530.829921	5125.079646	0	359.82	11.21	2.05	3.67	111.35	10.99	01/11/2013 10:00
530.829921	5125.079646	0	373.49	10.36	1.71	3.12	110.78	11.13	01/11/2013 10:10
530.829897	5125.081769	0	361.16	10.59	1.79	2.76	109.4	11.74	01/11/2013 10:20

Figure 3.2 CSV file content

3.3. User types

The users for this study are divided in three groups:

3.3.1. Common users

In this group, the people with no high geostatistical knowledge or model analysis skills are considered, like the following profiles:

Citizens: people interested in the processed nearly real time sensor network data like air quality for common reasons like planning outdoor activities; e.g. a sportsman concerned about air pollution.

Social groups: organizations interested in organizing outdoor activities, environmental or health issues, e.g. scouts groups, sport clubs.

3.3.2. Advanced users

Staff members of a university, organization or governmental institution; interested in the analysis and interpretation of results either for supporting decision making-processes or for educational purposes. Thus, they need specialized tools to determine the data distribution, predict values at unsampled locations and check the prediction's quality. The following profiles may be used to illustrate this group:

Researchers: people interested in the spatial prediction models and their quality assessment; e.g. university staff or sensor network administrators.

Service providers: institutions or agencies involved in the implementation of similar services and willing to integrate them; e.g. a similar project staff members.

Decision makers: authorities interested in the derived pollution map as well as its uncertainty to support their policy-making, local planning, health issues etc. e.g. local governments.

3.3.3. Developers

This group includes the information and communication technologies personnel, interested in the system interoperability and scalability, either for knowing technical specifications and requirements or for educational purposes. Thus, they need technical communication tools such as developer's guides, programmer manuals and standard modelling and representation elements like entity-relationship, use case and other UML diagrams. The following profiles may be used to illustrate this group:

System developers: personnel in charge of the future development and scalability of the platform; e.g. Eindhoven AiREAS project technic staff.

Service developers: technicians involved in new web services implementation, whose concern is the integration of these services; e.g. programmers from similar projects.

Service consumers: technicians interested in consuming the web processing services outcomes through APIs, etc.; e.g. similar client interfaces or mobile application developers.

3.4. Assumed scenario

The following examples illustrate the three different user type scenarios:

Remco van der Panne is an Eindhoven citizen who practices outdoor sports regularly. He wants to run 20 kilometres today but before starting he needs to plan the route, considering the air pollution levels throughout the city; he uses a mobile phone with internet access to obtain the last measurements from the AiREAS mobile app, so he can find out which areas of the city are healthier in terms of air quality. He gets the air quality map with a good-medium-bad scale for it, based on the World Health Organization (WHO) guidelines (WHO, 2014) and finally he can report the outdoor activity he is practicing today, if he wants to help the local authorities to improve their planning for sport facilities regarding location and open hours and the AiREAS management to determine where is this information required most often.

Hans van Gurp is an engineer working on system development and one of his duties is the opaque integration of new components to it, like a different type of sensor, a new client platform to deliver the measurements, etc., in such a way that interoperability and data integrity can be guaranteed. Thus, he will use the communication platform design to plan the new component's integration and establish the technical requirements that it has to fulfil before being part of the system.

Johannes Unglert is an AiREA's researcher who is in charge of the measurements' quality assessment, he needs to perform statistical and geostatistical processes on the platform to determine whether the data is normally distributed, to predict air pollution values at unsampled locations and check the prediction's quality. Thus, he uses a web browser to enter to the platform and get the corresponding data histogram as well as a graphical representation of the current kriging interpolations and the kriging variances, computed for every point within the city as well as diagnostic measurements derived from the cross validation method.

Figure 3.3 provides a graphical description of the workflow through the platform.

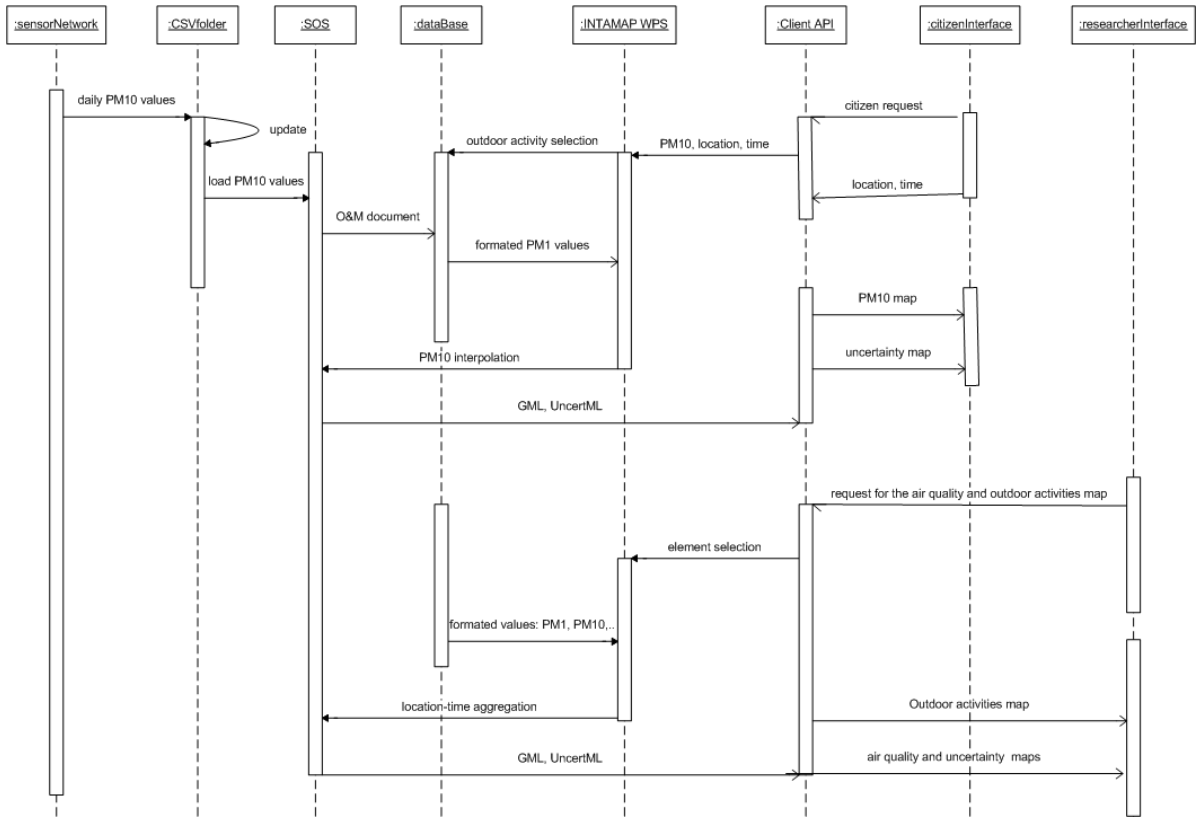


Figure 3.3 UML sequence diagram, system workflow

4. SERVICE-BASED APPROACH TO DEVELOP A SHARING AND PROCESSING SENSOR DATA PLATFORM

Since the first aim of this study is to determine the feasibility of using web services to facilitate the communication among the components involved in sharing and geostatistical processing of sensor data, it is necessary to analyse the different platforms and data models that are potentially suitable in terms of compatibility and taking the characteristics established in Chapter 2 into account, as well as the conditions needed to have the data in nearly-real time, which include the automation of such workflow by using complementary tasks.

4.1. Proposed realization of a platform for sharing and processing sensor data

The high level architecture of the proposed design for the service-based communication platform for sharing and geostatistical processing sensor data is shown in Figure 4.1; such design allows the data transfers throughout the different components involved from the sensors network to the web services, including the sensor observation service (SOS) and the web processing service (WPS), and also to the client applications. This chapter contains a description for each of these components and their complementary tasks.

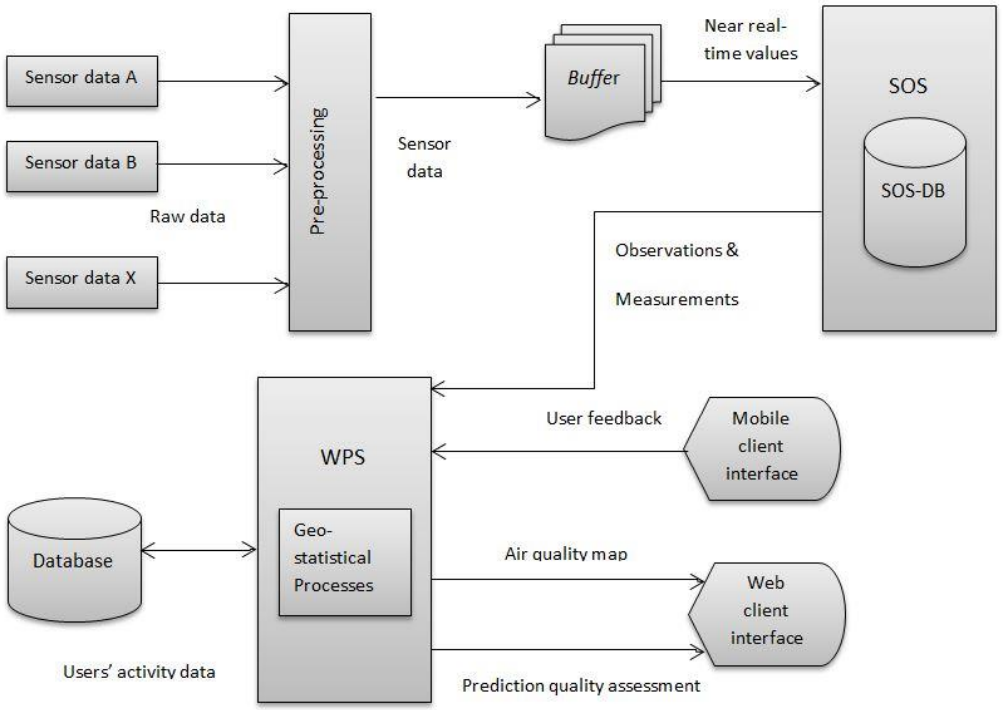


Figure 4.1 High level architecture

4.2. Pre-processing

The sensor data might be provided either in a standardized or in a specific format, defined by a sensor vendor. In the first case these data can be uploaded directly to the Sensor Observation Service (SOS); and in the second case they have to pass through a pre-processing stage in order to fulfil all the requirements for being represented in a standardized format.

Regarding the temporal availability of data, there are also two scenarios: either they are being provided by the sensor network in real time, in which case they can be used as input by the Web Processing Service (WPS), or they have to be prepared before starting the geostatistical processing. The pre-processing is described by figure 4.2.

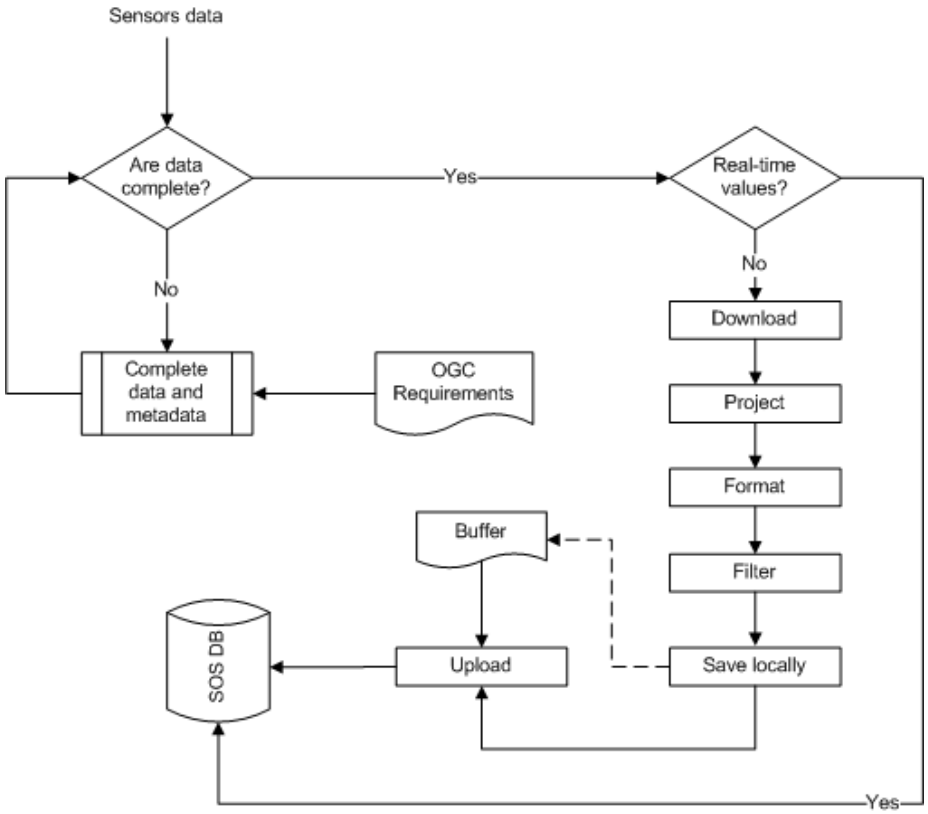


Figure 4.2 Data pre-processing workflow

This pre-processing includes the tasks that are necessary to prepare the raw data in such a way that they can be used as a proper input for the processing stage, as described below:

4.2.1. Raw data completeness

Before being used and processed through web services, data must have the required completeness and format compliance defined by the Sensor Web Enablement (SWE) in the Open Geospatial Consortium Inc. (OGC, 2015a), including their metadata and all the measurements description (attributes) that are compulsory to populate the Observations table in the SOS standard database (Figure 4.5).

4.2.2. Real-time measurements

Since environmental phenomena such as air pollution are dynamic, sensor data must be provided in real time, so they can be retrieved by the corresponding WPS from the 'buffer' (described below) or the SOS database (Figure 4.5) with the proper temporal resolution and considering their temporal validity, i.e. for how long may the measurements be considered valid. If this is the case the data can be processed directly without any additional treatment; otherwise additional pre-processing is required in order to have near real-time values that can be used as input for the geostatistical processing.

4.2.3. Buffer

It is also necessary to extract from all the historical records available in the sensor network, the last values needed for the geostatistical processing, i.e. last minute, hour, day or week depending on the temporal validity of data, which is included as an attribute in the SOS database 'Observation' table (Figure 4.7). These data is stored in a temporary data-storage structure that contains the current dataset, called buffer.

4.2.4. Data downloading

Before starting the pre-processing, the last valid dataset has to be available locally, this can be done by downloading it to the same hardware platform that is going to be used for the pre-processing stage; in this way the data transfer is done only once for this stage of the workflow and the network traffic does not suffer a high and continuous increase.

4.2.5. Data projection

Raw data contain the coordinates, in the columns 'east' and 'north' as numbers, as described in Figure 3.2; these values can be used to project the data by indicating the Coordinate Reference System (CRS) to the chosen software tool so it can create the geometry and the east and north values can be transformed to a spatial structure in order to locate the measurements which is a necessary condition to determine the spatial correlation among the data attributes.

4.2.6. Data formatting

Raw data retrieved from the sensor network might be in a wide diversity of formats; in the present case study as described in Section 3.2 (Data description) the archived data, is provided in hdf5 format (HDF-Group, 2014) commonly used for sensor data due to the high volume and growing rate of them; in a like manner, there available is a folder containing one file in Comma Separated Values (CSV) format for every monitoring station with a temporal resolution of ten minutes for the last-day measurements. Thus it is necessary to ensure that all the data models involved are standards compliant in order to guarantee the suitability of input and output data for each stage.

All of the data provided by the sensor network have to be converted in order to be suitable for the Sensor Observation Service (SOS), as follows (Na, et al., 2007):

- The observation, defined as the act of observing a phenomenon, needs to be described through a document based on the Observations and Measurements format which is the XML schema defined by OGC.
- The feature of interest, which is a real world entity targeted by an observation, has to be encoded in Geography Mark-up Language (GML).
- The sensor's metadata must be described by using the Sensor Model Language (SensorML).

4.2.7. Data filtering

Due to the fact that the monitoring stations ('Airboxes' in this case) are exposed to bad climate conditions, electrical energy issues, etc. some of the observations might be inconsistent or incomplete; and these missing or wrong values might cause an inconsistent outcome when performing the geostatistical

processing. Thus, it is necessary to filter the dataset in order to discriminate such data, or fix them if that is possible, by replacing the missing values with well-known and valid ones, e.g. the measurement location with the value of the previous observation taken by the same sensor or the current time for measurements whit some clock failure.

4.2.8. Data uploading

Once the dataset is pre-processed the structure containing these data (the buffer) needs to be uploaded to the corresponding sensor observation service (SOS) database tables, in order to allow the web processing service (WPS) to retrieve these data and start the geostatistical processing phase.

4.3. Sensor Observation Service

The SWE Sensor Observation Service (SOS) standard defines a web service interface for querying sensor data, including: observations, metadata and representations of observed features (OGC, 2015b), which is another requirement of this study; this is why, besides the aforementioned pre-processing a SOS component is included in the proposed communication platform architecture.

This standard also defines interoperable means for adding and removing sensors in a sensor network as well as the operations for inserting new sensor observations; providing standardized access to them and to the sensor descriptions and using the Observations & Measurements (O&M) standard to encode the sensor observations plus the Sensor Model Language (SensorML) to encode the sensor descriptions (Chu & Buyya, 2007). Figure 4.3 shows its general workflow.

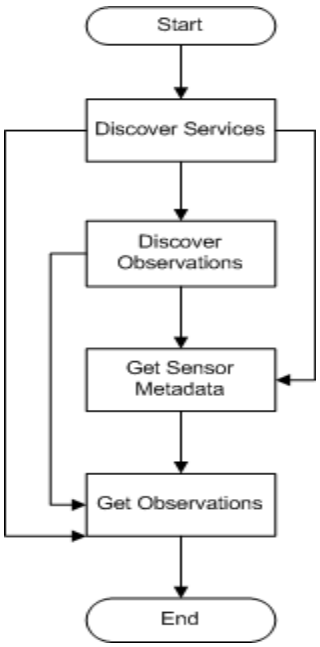


Figure 4.3 Sensor Data Consumption Workflow

The SOS approach consists in modelling sensor networks, sensors, and observations; covering all the sensor types and supporting all the user's requirements by using the standard properties of sensor data to

provide specialized operations for observations; thus, it can be applied to all domains using sensors to collect data, encapsulating the domain-specific details in a second layer, and allowing the observations to be processed through a generic client. An extract of this model is shown in Figure 4.4.
A SOS implementation is included in this design due to the fact that its capabilities are suitable to handle data with the characteristics described in Section 2.2, as it has strong features for storing big amounts of data as well as for querying on them. Furthermore, the SOS together with other OGC specifications, provide the interoperable capability for discovering, binding to and interrogating individual sensors, sensor platforms or sensor networks providing data in archived, real-time or simulated environments (Na, et al., 2007).

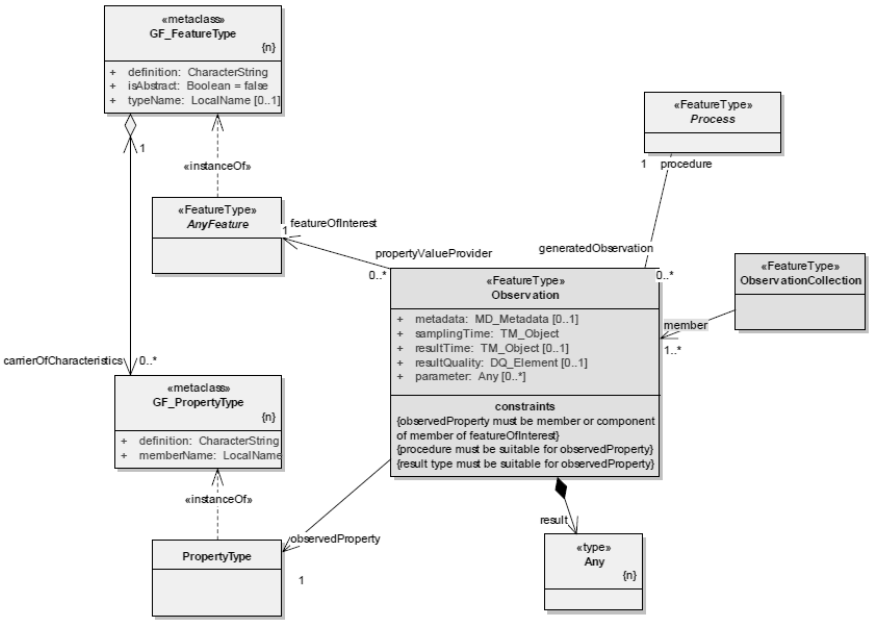


Figure 4.4 SOS model extract (OGC, 2015b)

.

SOS database

A generic database schema is needed for this project, so it can be used not only for a particular application case but for a wider range of domains (air quality among them). Thus, a model built-on the aforementioned O&M schema is considered suitable as it allows representing generic observations.

Since at this stage the 'buffer' data must fulfil the OGC O&M format specifications, the buffer content can be uploaded to the SOS database whose model is presented in Figure 4.5 with 'observation' as the central entity and the peripheral entities that store the sensor descriptions and metadata.

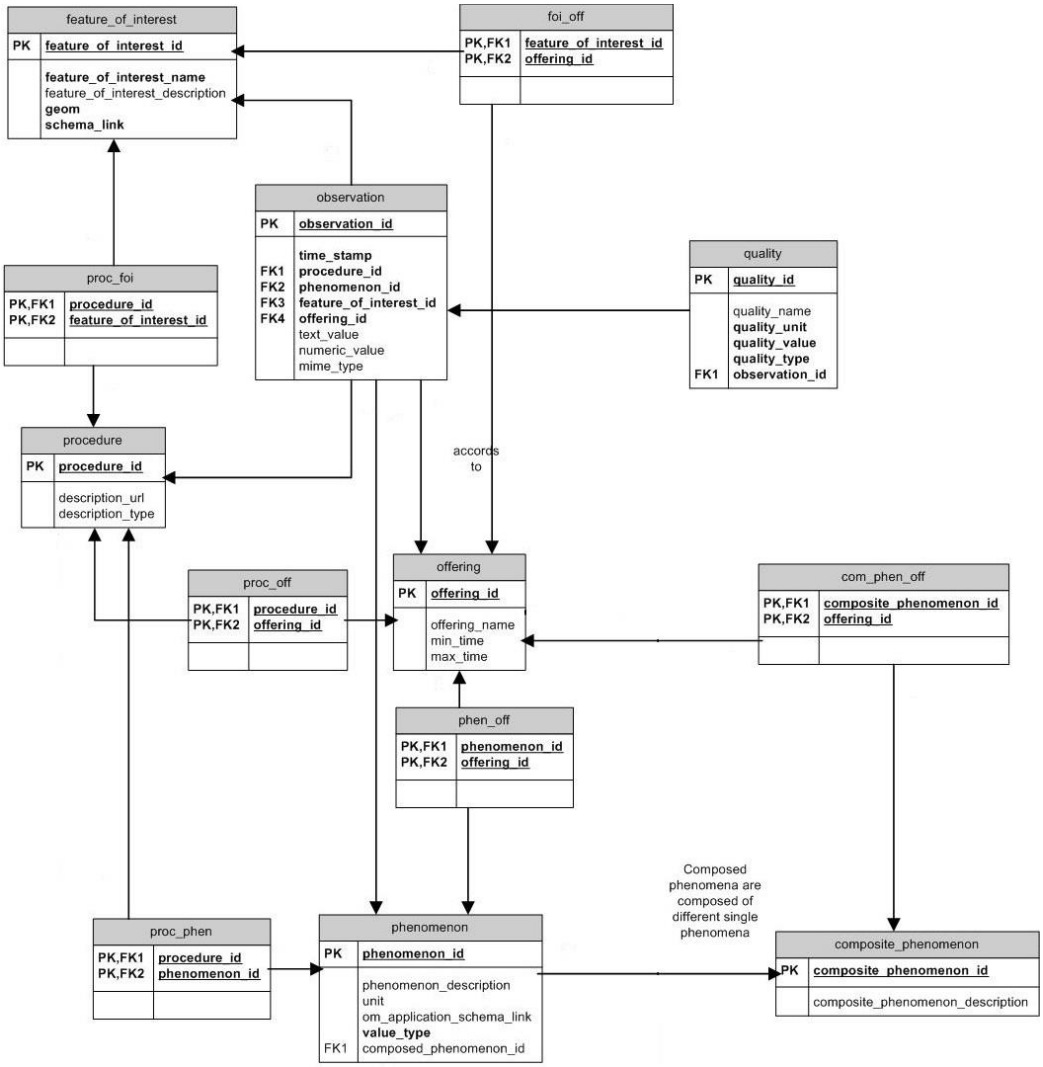


Figure 4.5 SOS database model (OGC, 2015b)

4.4. Web Processing Service

After ensuring that sensor data have proper formatting and availability for the processing stage another web service has to be implemented, to allow the creation of processes to run on the web service, including the performing of geospatial calculations or models plus making these processes available and readable by the different client platforms.

The OGC Web Processing Service (WPS) standard provides generic specifications for accessing, finding, and using geospatial processes (OGC, 2015d); according to these specifications, every process must be in a separate document in order to achieve far-reaching interoperability (Schut & Whiteside, 2007) as required for this study.

Furthermore, WPS specifications define input and output data generically in such a way that they may include image-data formats, data exchange standards, etc. and even calls to other OGC web services can be used as input data. These input data types are classified as follows:

- Complex data: including imagery, XML, CSV as well as proprietary or specific data structures.
- Literal data: like single numerical values or text strings.
- Bounding box data: like geographic coordinates for a rectangular area.

Regarding the output data types they can be grouped as follows:

- Raw format: when the WPS produces only one output and it is in the same format as the input. E.g. a buffered feature as a response to a request on the input feature for instance encoded in a jpeg format.
- XML-embedded: when the output requires the process results plus metadata about the request they can be wrapped in an XML response. This response might also contain references to web-accessible locations from which the outputs can be downloaded.
- Execute response location: for long-running processes, the WPS output may consist in a reference to a location where another execute-response is stored.

4.5. Geostatistical functions

Once the sensor data are properly formatted and available for processing, it is possible to address the second aim of this study: the sensors coverage issue. In this regard, geostatistical methods, are considered for this study as they have been applied in a variety of related disciplines (Li & Heap, 2008). Thus, spatial interpolation can be effectively used for estimating pollutant levels in areas with no measurements available (Singh, et al., 2011).

4.5.1. Variogram

As explained in Section 2.4.3 (Variogram), this function quantifies the assumption that things nearby tend to be more similar than things that are farther apart (W. R. Tobler, 1970 and W. Tobler, 2004); it is also a way to relate uncertainty with the distance between observations (Chilès & Delfiner, 2009)

The variogram can be obtained by plotting the experimental variogram values against distance classes, and the necessary input for kriging operations (e.g. type, sill, range and nugget), can be obtained by finding a model or function which fits these experimental variogram values; i.e. the experimental variogram can be used to construct a theoretical one by minimizing the variance and mean absolute error; then the semi-variance of a location within a given distance can be calculated (interpolated) through this theoretical variogram.

4.5.2. Ordinary kriging

Ordinary kriging, a stochastic and univariate method (see Section 2.4.1 – Geostatistical Models), is considered in this study as it is one of the most widely used interpolation methods described in Section 2.4.2 for environmental data (Webster & Oliver, 2007); and this selection also considers the fact that AiREAS project's dataset allows to use PM_{10} as primary variable and there is no other measurement provided which can be used as co-variable; besides, the obtained surface is needed in gradual fashion; and both deterministic and stochastic parts are required.

The suitability of this method is indicated in (Li & Heap, 2008) for cases when data seem to have a trend and they are sufficient to compute a variogram. Besides, it may use semi-variograms, which can be modelled from the data, to characterize autocorrelation or spatial dependence (Cressie, 1988).

4.5.3. Accuracy assessment

As established in Section 2.4.4, cross-validation is a method for testing the assumptions validity either on the model, e.g. the type of variogram and its parameters, the size of the kriging neighbourhood; or the data, e.g. the values that do not fit their neighbourhood like outliers or pointwise anomalies (Wackernagel, 2003). This method consists of the prediction of a value for an observed point, based on the rest of the points, and the comparison of the obtained value and the actual measurement; this procedure is applied to all the monitored points.

Usually, the result of kriging is the expected value (mean) and the variance computed for every point in a region. Kriging variance is the variance of the prediction errors (Gao, et al., 1996), which considers the kriging mean for every location as the average of the whole ensemble of possible realizations, conditioned on data; thus kriging variance is the variance of that ensemble.

4.6. Client side

In the pursuit of interoperability, the present design also considers exchangeable components in such a way that they can be replaced by a different but analogue 'part' to accomplish the same functionality. Interoperability can be achieved by giving developers the possibility of having their own implementation of a certain component.

One clear example of this property is the creation of an Application Programming Interface (API) to give developers the possibility of implementing a different system's front-end according to some new or simply different user requirements.

The Service Oriented Architecture (SOA) approach is the basis of this study, thus web services are being used to build the system components and standard protocols for facilitating the communication among them in a platform-independent way. The most accepted of these standard means to do so, are briefly described next:

4.6.1. Simple Object Access Protocol (SOAP)

Since it is XML-based it has a rather rigid-structure messages and as a consequence the need of parsing the XML response (Box et al., 2000). WSDL can be used here to specify the request parameters as well as for describing the expected result.

4.6.2. Representational Estate Transfer (REST)

This is not as rigid as the previous one as it allows to use not only XML but also Java Script Object Notation (JSON), and even plain text to represent the resources (Rodriguez, 2008). Besides, it uses standard Uniform Resource Identifiers (URI's) to invoke web services and the HTTP methods (GET, POST, etc.) to perform operations. However, it has the following constraints:

- Uniform client's interface for all resources.
- All requests must carry session-oriented information.
- It must allow the clients or intermediaries cache responses marked by servers.

The next step is to use a Hypertext Transfer Protocol (HTTP) client to invoke the developed web API; it can be achieved, for instance by using JavaScript and jQuery or through a web browser; this client receives the requests and route it to an action which can be either an API's standard or custom method. Hence, the present work considers a web interface as the proposed means for sharing the outcomes with the different type of users described in Section 3.3.

5. PROTOTYPE IMPLEMENTATION

This chapter contains a technical description of the various tools and resources utilized for a prototype's implementation based on the design presented in Chapter 4; following an incremental - iterative approach (Larman & Basili, 2003); in which the different components are developed at various times or rates and then integrated, resulting in the incremental addition of features for each cycle until the system is complete. Figure 5.1 shows the planned project workflow with all the increments and iterations involved.

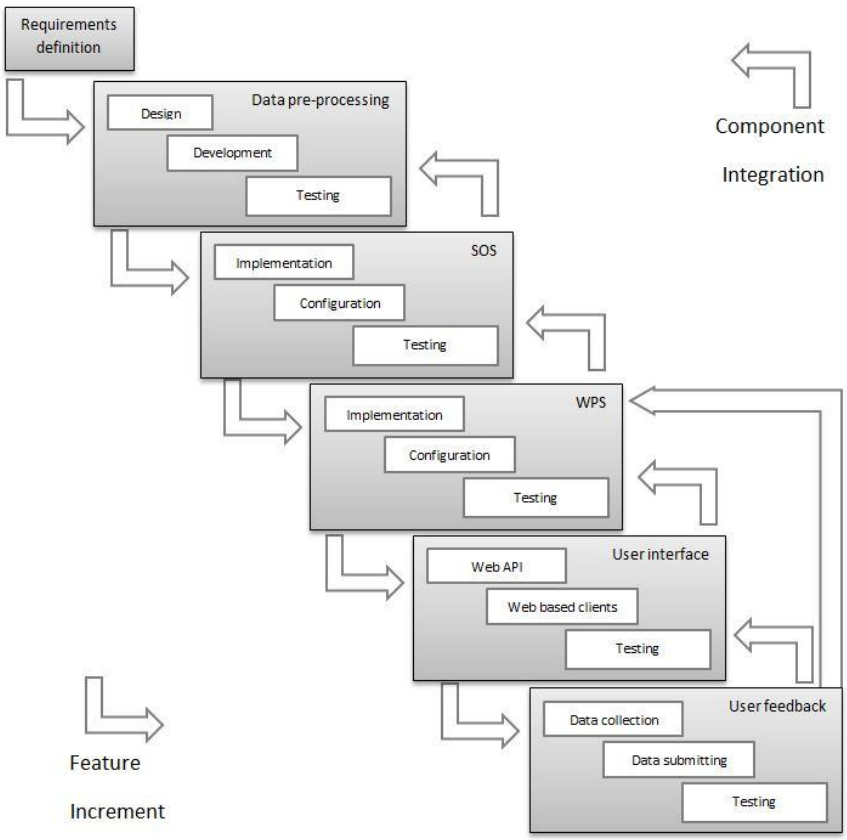


Figure 5.1 Incremental – Iterative workflow

The prototype implementation plan is composed of six main tasks described in Figure 5.1; though, only the first four were completed for this work due to time limitations, the fifth component is partially implemented (the 'advanced user' interface) in order to give a general description of how these users can interact with the platform.

5.1. Selection of technical resources and tools

Several resources and tools have been evaluated during this implementation, including standards compliant service-based solutions for each phase and programming languages and their own libraries for

the complementary tasks. All the tested and selected resources for each stage and component of the platform are described in this chapter.

Python has been chosen for the pre-processing tasks, because is a general purpose programming language, which is easy to learn (Python Software Foundation, 2015); plus, there is enough on-line support and developed libraries to accomplish the planned targets for the present work.

In a like manner, R is used for the geostatistical functions implementations as is a free language for statistical computing and graphics, which compiles and runs on a wide variety of platforms such as Unix, Windows, Linux, etc. (Institute for Statistics and Mathematics, 2015).

Besides, both Python and R are capable of handling the most widely used data formats and they are being updated constantly by adding specialized libraries for new data models, which is relevant on this platform as it considers future integration of components.

Regarding the required databases implementations, both the SOS database and the system database (see Figure 4.1) are built on PostgreSQL, an open source object-relational database system which runs on the main operating systems; supports all the features required for this study (Section 2.6); and is entirely standards compliant (The PostgreSQL Global Development Group, 2015).

Before installing the web services, the Apache Tomcat (Apache-Software-Foundation, 2015) was installed to be used as the web server and servlets repository on which those services are mounted. Once it was installed, configured and it is running on a server, the 'tomcat manager' interface was used to configure the following components.

Since, at least two web services, the SOS and the WPS, are included in the presented design; 52° North implementations are selected as the basis for the present platform, because they are fully compatible and interoperable as they were planned and developed this way (52°North-Initiative, 2015a).

Nevertheless, two or more alternative realizations have been tested for every stage, in order to check out the interoperability among components; and they are all described in the corresponding chapter sections. It is also important to mention at this point that such tests are not aiming to measure an efficiency index, as it is none of this study's objectives, but only to establish whether a component can be replaced or not in such a way that the whole platform still works properly.

5.2. Technical setup

5.2.1. Pre-processing

The target in this stage is the preparation of data in such a way that they become suitable for geostatistical processing, this is divided in sequential steps and each of them is meant to be adaptable or upgradeable and the order in which they are executed may be different, depending on the initial conditions in which the raw data is provided, as described in Section 4.2.

Assuming that the dataset is complete as described in Section 4.2.1 and the sensor measurements are not been provided in real time (Section 4.2.2), the pre-processing stage is required and its implementation is composed of the following elements and tasks:

Buffer
Such structure is needed to temporarily store data while moving them between processes, as established in Section 4.2.3, it always contains the currently valid dataset, e.g. last measurements, projected or filtered records, and so on.

In the present case, as the raw data are provided in CSV format and both Python and R have special libraries and functions (e.g. Python: csv library, R: read.csv and write.csv functions) for handling this type of files it is not necessary to change the original format for the buffer implementation.

An alternative implementation of the buffer is to use a database table to play this role; although, this way of doing it generates additional data transfers e.g. from a file to a table and so on.

Data downloading
Most likely, the data to be used are not in the same hardware platform on which the web services are running; then, in order to avoid network traffic increments, the dataset is automatically downloaded by means of a Python script through the available libraries for handling data of such nature.

For the present case study: htmllib, formatter, urllib, sys, getopt, os and psycopg2 Python libraries were utilized; as the raw data are provided through a public server containing a folder with one CSV file for every sensor as described in Section 3.2 (Data description), each of these files contains all the measurements collected by the specified sensor every ten minutes and the last twenty four hours measurements are appended daily.

Then, the script parses the html document (web page) looking for hyperlinks; it creates a list of the found links and then it gets each link's content, one by one (see Figure 5.2). After running this script all the CSV files are stored locally in a folder called "data".

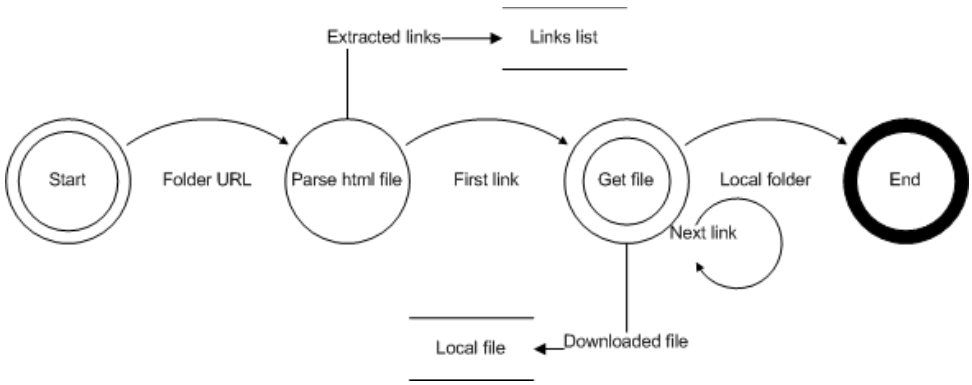


Figure 5.2 Data downloading procedure

Data Projection
It is also necessary to convert the raw data coordinates to the proper values (Section 4.2.5). This study's data are coded as follows: the first two numbers to the left of the decimal points are minutes of a degree, and the two figures to the left of that are the degrees. It is therefore necessary to convert these raw data to

decimal degrees (longitude, latitude) and then convert these values to the Dutch coordinate system, i.e. Rijksdriehoek (RDH) by using the "sp" and "rgdal" R libraries and the "spTransform" R function.

Data formatting
Basically the required data type transformations for this case are from CVS to an XML Schema-based format; to do this the Python libraries 'sys', 'os' and 'glob' were used and, again every file is parsed in order to build new XML files based on the corresponding schema, as mentioned in Section 4.2.6 (Data formatting).

Data Filtering
There are three important issues with the provided dataset that have to be solved during this stage:

- Some of the 'Airboxes' are located outside the city and they have to be discriminated in the first place, as these measurements have a negative effect on the quality of predictions because of the low spatial correlation they have with the rest of the data, and also because they are no within the defined study area. This is done by using the 'read.csv.sql' function from the 'sqldf' R library; the R 'subset' function is another way for achieving such data filtering.
- Every ten-minutes each 'Airbox' delivers the GPS location values together with the rest of the measurements, these locations however are not exactly the same for each record (because of the changing satellite positions across their orbits), thus, the sensors' measurements and locations (tables 'measurement' and 'sensor' respectively) are stored separately so each sensor location is established only once and the set of measurements are related to their corresponding sensor through the sensor Id by setting a foreign key constraint.
- The time given by each sensor's internal clock is not exactly the same for all of the measurements (it differs for no more than ten minutes though); thus, it is necessary to set an unique time for all the collected values at a certain moment; this is done by getting the system's current time and then inserting this value into the time attribute for each measurement.

Data Uploading
This last step before starting the actual processing of data is also realized automatically by using the 'psycopg2' Python library's functions to perform SQL queries on the database tables.

5.2.2. Sensor Observation Service

Two different implementations were tested for this service: Map Server for Windows (MS4W) SOS and 52° North (52N) SOS, since they both are OGC standards compliant there are no relevant differences, regarding interoperability. Thus, 52N SOS was selected due to its full and guaranteed compatibility with the WPS component as they both are developed and maintained by 52° North (52°North-Initiative, 2015a).

The SOS provides an Application Programming Interface (API) for managing deployed sensors and retrieving sensor data; it has three core operations, which produce different standards compliant outputs as shown in Figure 5.3, they all are XML schema based specifications, defined by the OGC.

- GetCapabilities which provides means to access the SOS service metadata.
- GetObservation, providing access to the sensor observations and measurement through spatiotemporal queries.
- DescribeSensor which retrieves information about the sensors and the processes that are generating those measurements.

There are also several options, according to the standard's specification, that are not described in this work, because the three above mentioned operations are enough to address the requirements defined for this study.
An alternative means for retrieving and filtering SOS data is the R package called 'sos4R' which also provides functions for transformation of data (results) and querying on the service metadata (Nüst, et al., 2011); the output has attributed columns for metadata and is fixed to a standard 'data.frame' structure.

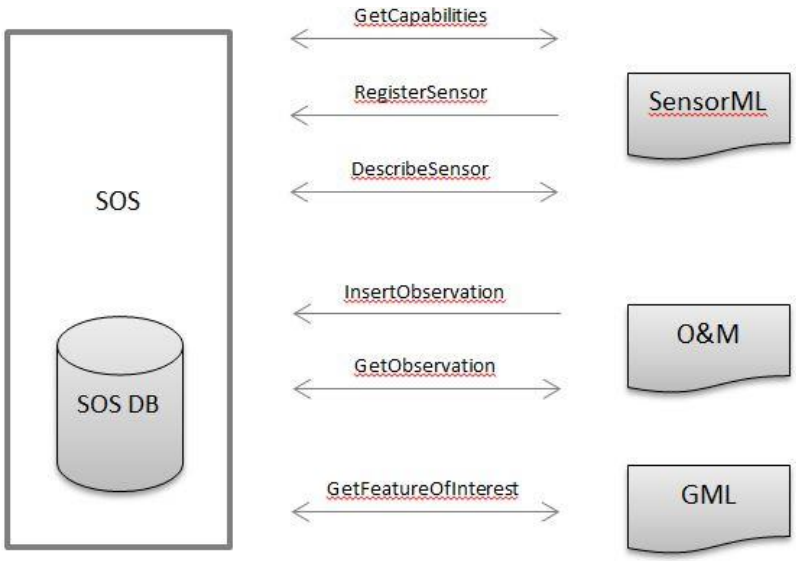


Figure 5.3 SOS specifications for operations and formats

The SOS database was implemented by executing the PostgreSQL script that is provided through the 'Administration' console; and the 'sos' database and all its tables are now available on the indicated host as shown in Figure 5.5.

It is important to mention that a simplified version of this database (see Figure 5.4) has been created for this prototype, because there is no metadata available for the provided dataset (from the 'AiREAS project'), which is necessary to populate some of the tables' attributes that are compulsory.

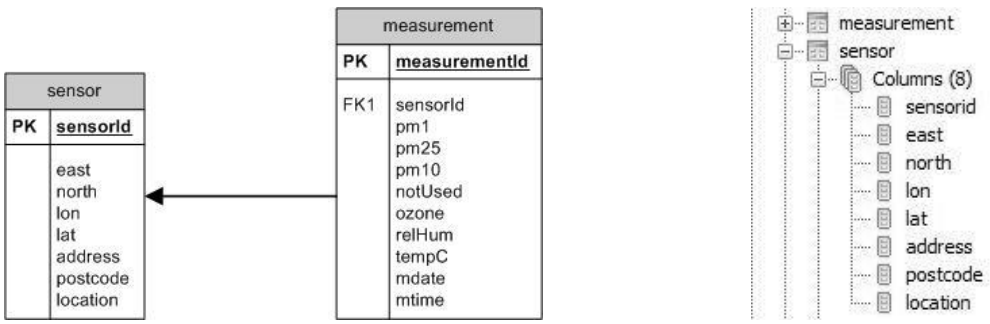


Figure 5.4 simplified version of SOS DB for this prototype

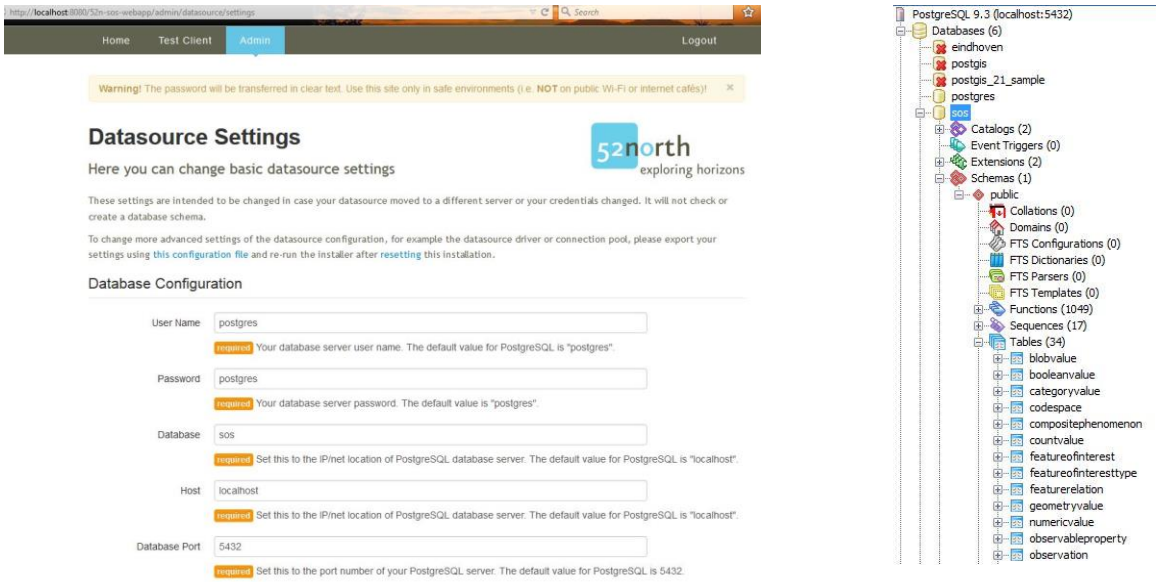


Figure 5.5 SOS database configuration console and implementation

5.2.3. Web Processing Service

Since the main purpose of this component is to facilitate the geostatistical processing of sensor data, specifically for interpolation as explained in Section 2.4.2; to get the data from the aforementioned SOS; and also to deliver the predictions as well as their variance (Section 2.4.4) to the users.

The first implementation was done by means of the European Commission project called 'Interoperability and Automated Mapping' (INTAMAP) (EC, 2009); providing several geostatistical interpolation methods through an OGC-based WPS; the Tomcat Installation console is shown in Figure 5.6. Besides, it has several client applications, including a Windows app called SeeSharp (Figure 5.7) which allows the interaction with 'INTAMAP' WPS.

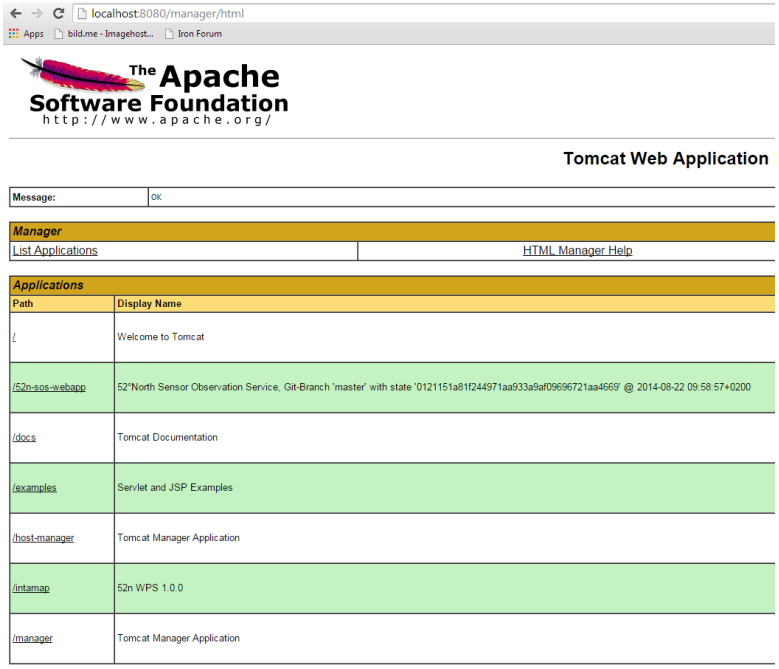


Figure 5.6 Apache Tomcat 6 Administration console to deploy 'INTAMAP' on the server

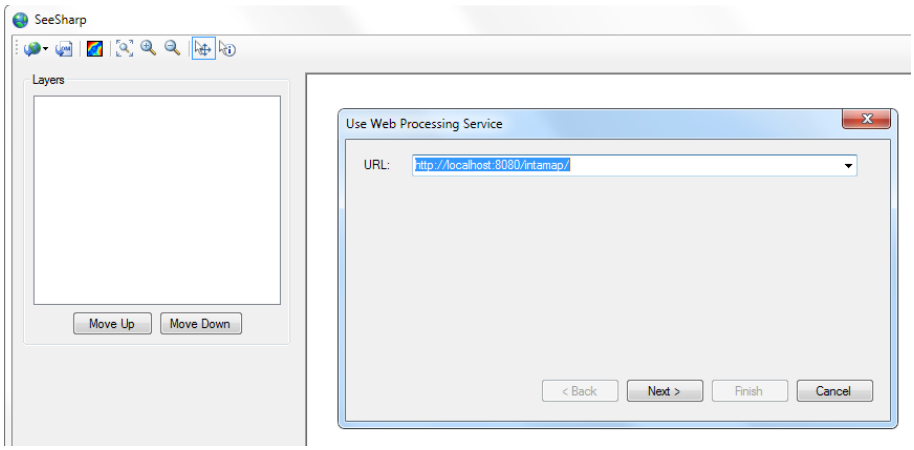


Figure 5.7 SeeSharp interface to set up the 'INTAMAP' WPS on the client side

INTAMAP's input can be configured to be retrieved, among others, from: a SOS, a Web Feature Service (WFS) and a standard XML file. The predictions (implemented by coding the corresponding R scripts) are delivered through a Web Map Service (WMS) and their uncertainty through UncertML (See Section 4.2.6).

The second implementation was done by using 52° North WPS4R (52°North-Initiative, 2015b), which is not only able to run processes coded in R but also in Java, so it is more generic. The implementation procedure is practically the same as the one described for R, by using Tomcat to deploy it, Figure 5.8 shows its interface.

Yet another implementation was tested for this study: pyWPS (DBU, 2009) is also standard based but is specially made for running Python code on it (Figure 5.8); even though running R scripts is also possible. These last two implementations can handle the same input and output as the ones described for 'INTAMAP'.

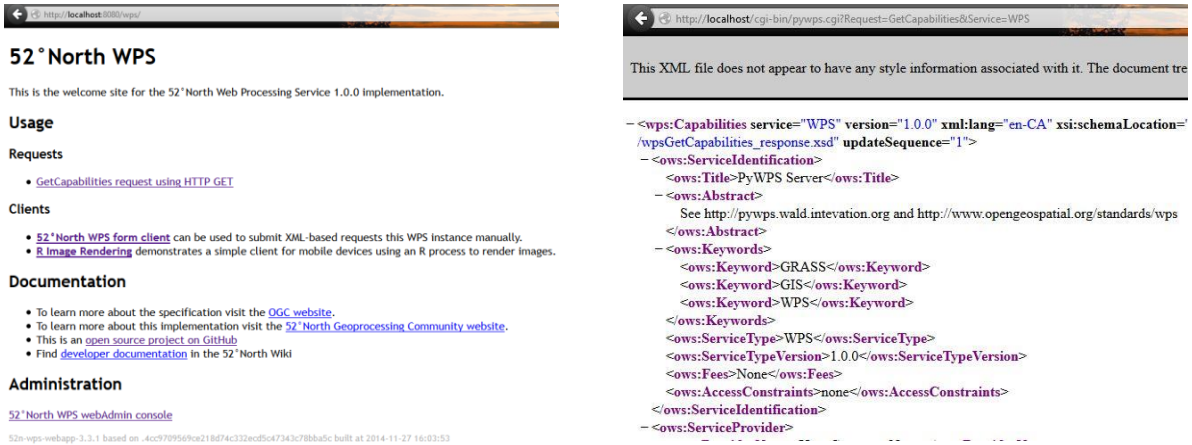


Figure 5.8 WPS4R (left) and pyWPS (right) 'GetCapabilities' requests

5.2.4. Geostatistical functions

This component contains the R code to perform the spatial predictions; the scripts are running on the aforementioned WPS implementations. Additionally, automating the process is required; thus, at this point the generation of the empirical variogram (see Sections 2.4.3 and 4.5.1) as well as fitting this variogram to the model was an important obstacle for the whole process' automation.

In this regard, two implementations have been selected as they can generate the variogram automatically, in such a way that the process can continue without human intervention. These are the R packages called 'intamap' and 'automap' as well as the functions described below.

The code for these two implementations to run on a WPS is the same as if they were to run on a desktop environment, except that at the beginning of the code they have to include some metadata: a description, an abstract, as well as some parametrized descriptions of the input and output (Figure 5.9). Once they include these metadata, they are placed on the WPS to be executed.

```
##### Performing ordinary kriging on the last 10 min measurements
# Author: Alfredo Vasquez
#wps.des: title = OK in R,
#abstract = predicting PM10 on unsampled points;
#wps.in: day, type = integer, abstract = Points for OK, value=6;
#wps.out: output ,type =geotiff_image, abstract =pm10 map;
## begin ####
require(gstat)
library(raster)
```

Figure 5.9 Metadata required for running R scripts on a WPS

'Automap' uses a function called 'autoKrige' and also some of the 'gstat' library functions for the interpolation process, except for the automatic generation of the variogram; it uses the following initial values in a method called 'autofitVariogram':

- Cut off: 0.35 times the maximum distance
- Range: 0.1 times the area diameter
- Nugget: the lowest semi-variance found in the sample variogram
- Sill: is the mean of the maximum and the median semi-variance value in the sample variogram
- Model: the one with the minimum sum of square error

On the other hand, 'intamap' uses the 'estimateParameters' function which is a wrapper around different methods for estimating the correlation parameters, including 'autofitVariogram' and 'copulaEstimation'; in order to generate the variogram after automatically selecting one of the available methods (e.g. inverse distance weight, automap, copula, trans-gaussian, etc.) and using this variogram as a parameter in the function called 'interpolate' to perform the spatial predictions.

The 'krige.cv' function, from 'gstat' library, is used to obtain the cross-validation and check the 'automap"s ordinary kriging success; by using the same variogram model and predicting each point from the rest of them. The diagnostic measures are: the Mean Error (ME) which indicates the bias; the Root Mean Square Error (RMSE) which indicates precision; and the Mean Squared Deviation Ratio (MSDR) which represents that cross-validation residuals should be equal to the prediction errors at each point, otherwise it would imply that the predictions are less variable than reality (Rossiter, 2012) as expected, because kriging is a smoothing estimator. Finally, by obtaining the absolute value of the cross-validations residuals and comparing them (or computing the difference) it can be determined which one has more accurate predictions.

5.3. Results

Since the present work consist of a platform design which facilitates the sensor data sharing and processing, the results are divided in three groups as follows:

5.3.1. Pre-processing

The pre-processing result is the automatic SOS database population; the raw dataset goes through all the pre-processing phases, and at the end of the 'uploading' phase, the last version of the 'buffer', as shown in Figure 4.2, is inserted into the corresponding tables (see Figure 5.5); such pre-processing automation is required in order to provided nearly real-time measurements and spatial interpolations.

The table 'measurement' (Figure 5.10) records the values used to perform the spatial predictions at a certain time, e.g. every ten minutes. It is useful because on one hand, these values may be retrieved at any other time to perform their corresponding interpolation once again, if it is required; and on the other hand, they may be aggregated to the previous or the next measurements to form for instance hourly measurements and improve the interpolations' input quality.

sensorid integer	pm1ugperm3 real	pm25ugperm3 real	pm10ugperm3 real	notused real	ozonugperm3 real	relhumpct real	tempc real	mdate date	mtime time with
1	8	11	17	0	5.5	107.73	-2.57	2015-02-03	05:32:17
1	4	5	9	0	19.9	64.12	7.11	2015-02-03	15:08:06
1	13	16	20	0	17.7	119.74	-3.83	2015-02-03	00:18:18
1	13	16	20	0	17.7	119.74	-3.83	2015-02-03	04:49:31
1	10	14	19	0	5.6	122.6	-4.32	2015-02-03	04:49:51
2	10	14	21	0	3.8	109.54	2.54	2015-02-03	05:32:17
2	16	20	27	0	12.7	107.51	2.98	2015-02-03	00:18:18
2	7	10	13	0	5.3	101.09	4.51	2015-02-03	15:08:06
2	16	20	27	0	12.7	107.51	2.98	2015-02-03	04:49:31
2	13	17	29	0	4.1	110.83	2.57	2015-02-03	04:49:51
3	12	15	32	0	7.3	111.38	3.45	2015-02-03	04:49:51

Figure 5.10 Table 'measurement', which is automatically populated

5.3.2. Service-based implementation

The part of this platform that allows the sharing and geostatistical processing of sensor data is composed of the SOS and WPS implementations, and the web client interface. These components are built on the aforementioned OGC standards and specifications, as they are required to be platform-independent, i.e. accessible with no restricted specifications or special software products on the client side.

Interoperability is pursued by following the Service Oriented Architecture (SOA) guidelines, which basically means that this platform is formed of a collection of standards compliant web service implementations that allow the clients, over an intranet or internet, to make requests for specific resources such as data or metadata and also to remotely invoke a software function by sending the call and some parameters to the service, then running the code on the server and sending the result back to the client.

As this study is also aiming to determine the feasibility of using the aforementioned web services to facilitate the sensor data sharing and processing, the implementation of an interoperable collection of such services is considered the result for this phase. The figures below display these components implementations: Figure 5.11 shows the implemented SOS configuration and testing console and Figure 5.12 shows the WPS administration console.

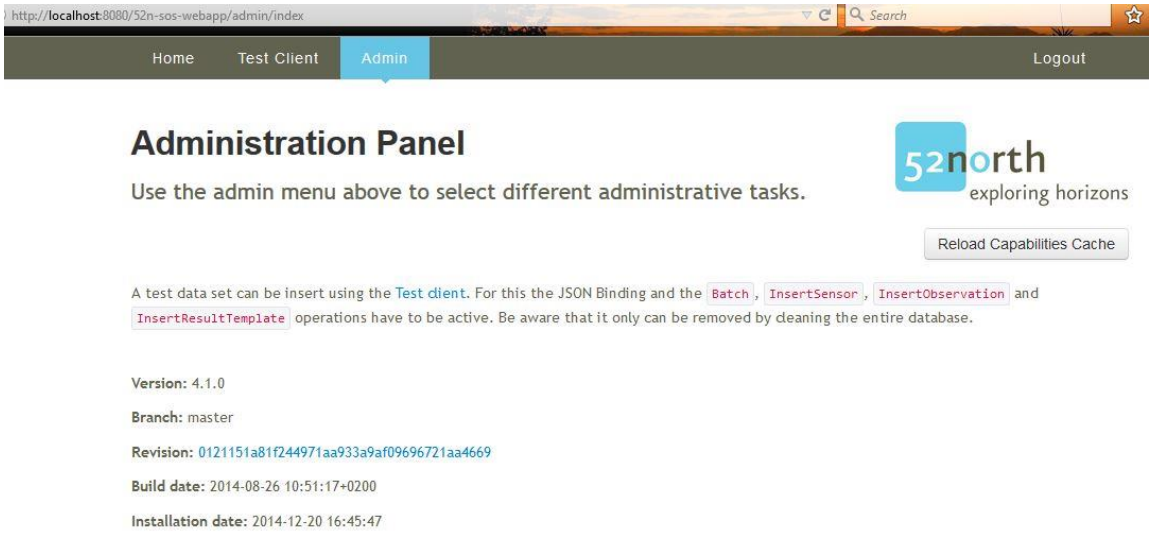


Figure 5.11 Implemented 52o North SOS console

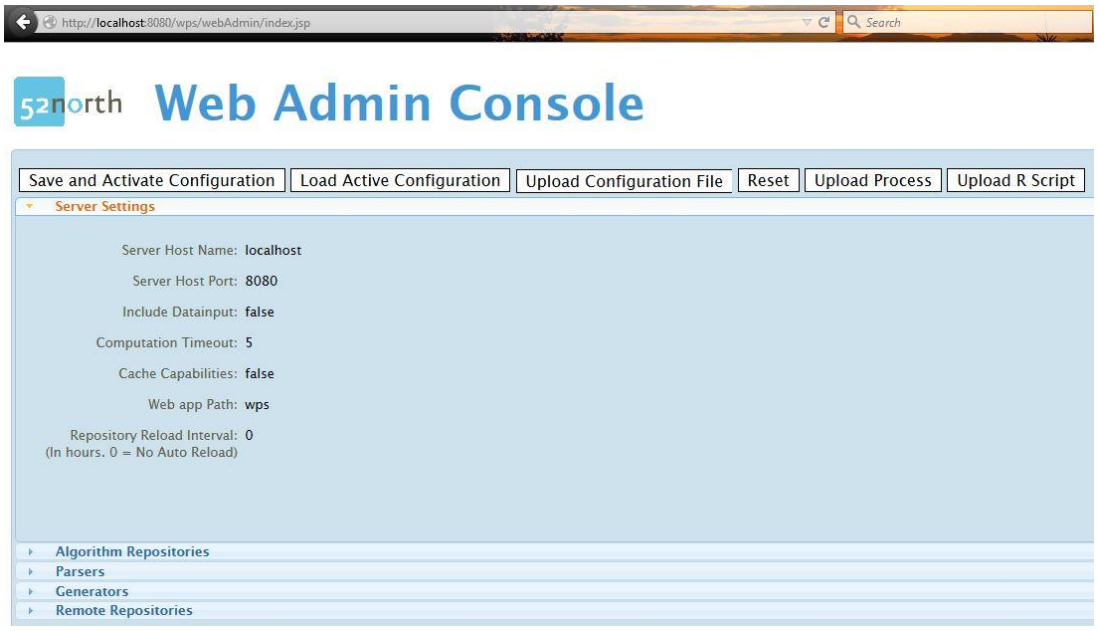


Figure 5.12 Implemented 52o North WPS4R console

The last component implemented for this prototype is the web client interface, which was built on the Shiny web application framework for R (RStudio, 2015). This tool was selected because it is fully R compatible and also because it can be learned quickly and it allows developing a basic web interface in relatively short time.

As explained in this chapter's introduction it was not possible to develop neither the 'common users' interface nor the user feedback component, due to time limitations. Figure 5.13 shows the 'advanced users' web interface (described in Section 3.4.2) which allows the user to select a library's set of functions (Section 5.2.4 Geostatistical functions) to perform spatial predictions and the element to be interpolated, as well as the automatically produced outcome: the experimental variogram, the fitted variogram model, the predictions and the corresponding standard errors.

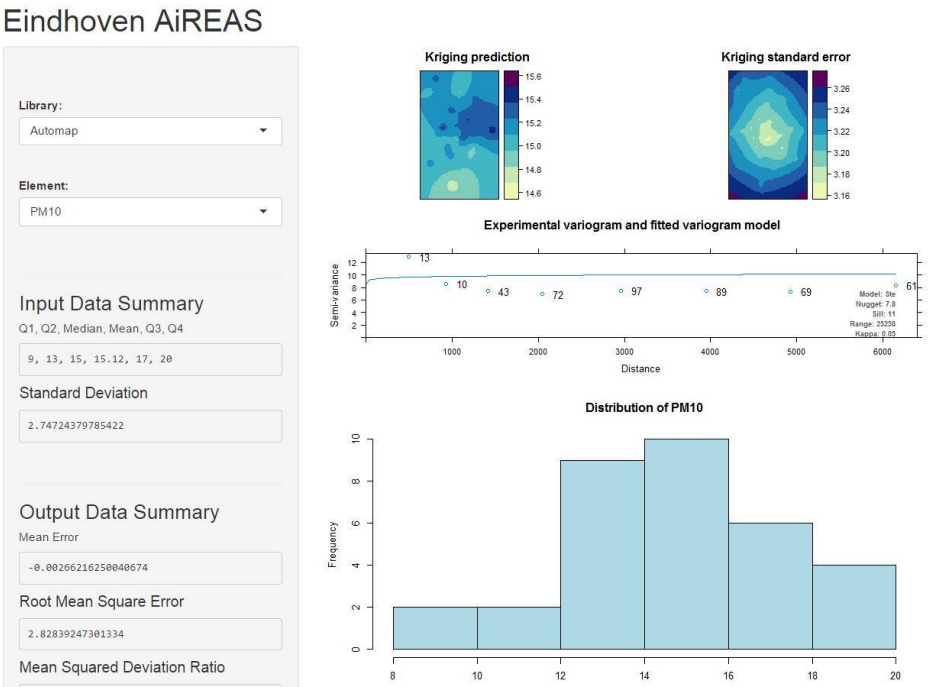


Figure 5.13 Implemented web 'advanced user' interface

5.3.3. Geostatistical processing functions

The data described in Section 3.2 were pre-processed and used for the geostatistical processing stage. Specifically, the measurements were delivered by the sensor network and are available on the SOS database to be the consumed by the WPS (see Figure 5.10).

The input dataset is composed of the last ten-minute's measurements, and the values from February the 6th, 2015 at 07:30 are used to illustrate the input for this study. Figure 5.14 shows the distributions of PM10 concentrations and the logarithmic transformation of the values and Table 5.1 presents the dataset summary.

The first histogram shows that the dataset has an approximately normal distribution and the second one shows that the log transform does not produce a significant improvement on it. Thus the PM10 values are used as input with no transformations.

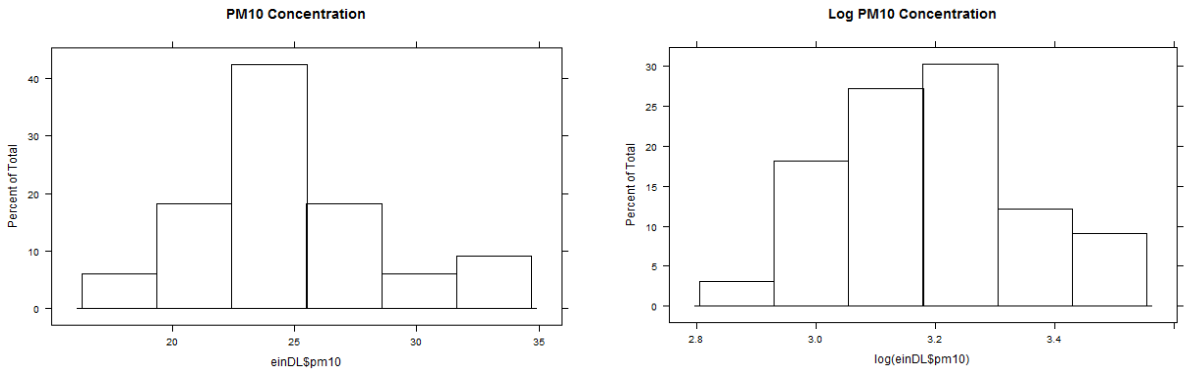


Figure 5.14 Histogram of the PM10 values (left) and the logarithm transformed values (right)

Min	1st. Qu.	Median	Mean	3rd. Qu.	Max	Std. Dev.
17.00	23.00	25.00	24.76	27.00	34.00	3.881288

Table 5.1 Input dataset's summary

As established in Research Objective 3, the next step is to automatically perform the spatial predictions. This was implemented, as described in Section 5.2.4, by using two different R libraries containing geostatistical functions: 'automap' and 'intamap'.

Automap

The 'automap' library uses a function called 'autoKrige' (see Section 5.2.4) to perform the ordinary kriging method on the input dataset. This is completed after 'autoKrige' automatically generates the experimental variogram plus the fitted variogram model, as shown in figure 5.15 (distances are in meters), and the estimated parameters for the model are in Table 5.1.

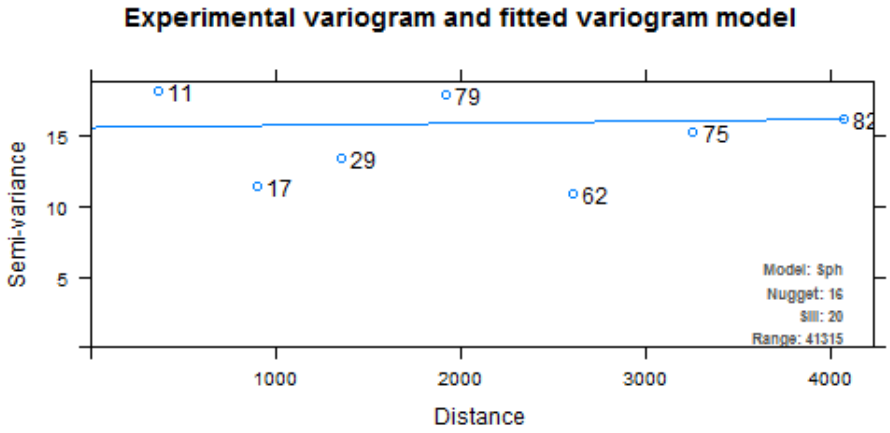


Figure 5.15 Experimental variogram and fitted variogram model generated by 'automap'

	Model	Psill	Range
1	Nug	15.639193	0.00
2	Sph	4.043014	41315.27
MinNP: 11		SSErr: 0.001315049	

Table 5.2 Model's parameters estimated by 'automap'

In this case, the sample variogram (Figure 5.15) does not show a clear spatial structure, this fact leads to the very long range required for the fitted variogram model (Table 5.2).
The 'autoKrige' function's outcomes for prediction and standard error may be plotted as shown in Figure 5.16, displaying the predicted mean concentrations of PM_{10} particles per cubic meter of air volume (m^3).

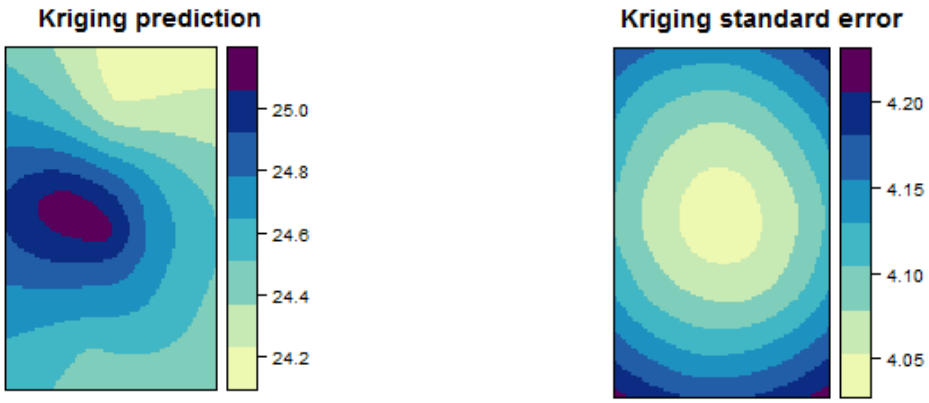


Figure 5.16 Plots of kriging predictions (left) and kriging standard error performed by 'automap'

Intamap

'Intamap' is the second library used in this work to predict values for the non-sampled locations, with the same input described in Figure 5.14 and the function called 'interpolate' as established in Section 5.2.4. The Experimental variogram generated by 'intamap' is shown in Figure 5.17 (distances are in meters) as well as the fitted variogram model.

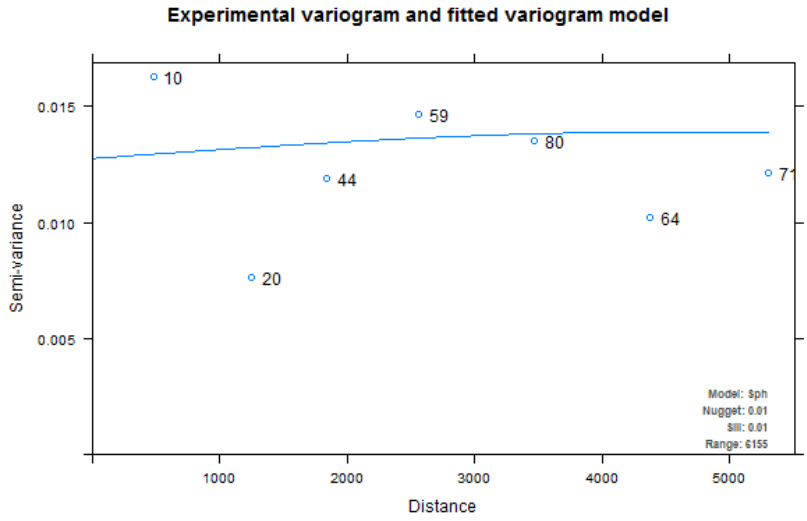


Figure 5.17 Experimental variogram and fitted variogram model generated by 'intamap'

	Model	Psill	Range
1	Nug	0.012754366	0.00
2	Sph	0.001130895	6155.194
Min. NP = 11			

Table 5.3 Model's parameters estimated by 'intamap'

Again, the variogram does not show a clear spatial structure (Figure 5.17) and the estimated parameters for the model (Table 5.3) display how these values differ from 'automap"s results because these operations are built on different algorithms and functions as explained in Section 5.2.4. In this case, 'intamap' is using the 'transGaussian' method which applies Gaussian transformations to the input data; this is why the semi-variance scale is also different.

The 'interpolate' function's outcomes for prediction and standard error may be plotted as shown in Figure 5.18 displaying the predicted mean concentrations of PM_{10} particles per cubic meter of air volume (m^3).

Figure 5.18 Plots of kriging predictions (left) and kriging variances performed by 'intamap'

Validation

Cross-validation, as explained in section 5.2.4, was performed by means of the 'krige.cv' function and the obtained measures, shown in Table 5.4, are used to compare 'automap' and 'intamap' results.

	'automap'	'intamap'
ME	-0.02317754	-0.02178094
RMSE	3.88857	3.746012
MSDR	0.9116817	1006.593
Kriging Variance (mean)	16.86	14.60

Table 5.4 Cross-validation diagnostic measurements

'Automap': the Mean Error (ME) (the bias indicator) is as expected almost equal to 0, the Root Mean Squared Error (RMSE) (the overall precision) is expected to be a short value (Section 2.4.4, Accuracy assessment) and the Mean Squared Deviation Ratio (MSDR) is nearly 1 indicating that for every point the residuals from cross-validation are similar to the prediction error.

'Intamap': the ME and RMSE values are very similar to the 'automap' values meaning a low bias (a bit lower for 'intamap'). Nevertheless, the very high value of MSDR indicates that the variability of 'intamap' predictions is being underestimated; and the kriging variance mean which is an estimation of all the prediction errors is also smaller for 'intamap'.

Finally, the 33 cross-validation residuals were compared as explained in Section 5.2.4 and the outcome indicates that there are 19 of them for which the absolute value of the residuals are less with 'intamap' than with 'automap' and 14 for which it is the opposite. Therefore, around 58% of the individual cross-validation predictions are better with 'intamap', this fact has to be considered when establishing which one is the most suitable function to perform these interpolations (addressed by Research Question 2.1).

6. DISCUSSION

An operational prototype of the proposed platform described in Section 4.1, has been realized and the results presented in Section 5.3 are discussed in this chapter in order to address the research questions that have motivated this study.

This discussion is divided into three sections which aim to report on the different activities that have been carried out as well as to provide further description of the sensor data pre-processing (Sections 4.2 and 5.2), the required service-based implementations for the prototype (Sections 4.3, 4.4 and 5.2) and the geostatistical processing functions that are required in order to perform spatial predictions (Sections 2.4, 4.5 and 5.2).

6.1. Pre-processing

This component, as described in Section 4.2 is composed of a set of tasks implemented through Python, R and SQL coding as described in Section 5.2 in such a way that the raw data, provided for the present case study (Section 3.2), have been successfully converted into a dataset that can be used as input for the prototype's processing stage.

This stage has been automated in such a way that a pre-processed dataset may represent nearly real-time values, as each task tackles specific raw data issues (Figure 4.2). For instance, in case it is necessary to handle missing or wrong values on the fly (see Section 4.2.7), e.g. when having a wrong time value due to a sensor's internal clock failure, a valid time can still be derived from the rest of the sensors or from the system's current time as it is known that it cannot differ from the actual measurement time more than ten minutes as mentioned in Section 5.2.1 (Filtering).

Moreover, the pre-processing stage can be generalized and applied to sensors raw data, as long as they are provided in a format supported by one of the programming languages described above or some other with specific libraries to handle such format. Besides, the tasks sequence can be adapted when necessary for a specific case, i.e. certain tasks might be required to be realized in a different order than the one described in Figure 4.2, except for the first (downloading) and last (uploading) tasks.

Nevertheless, the lack of data completeness and metadata, as in the present case study's dataset, was an enormous obstacle for the proper utilization of data on the corresponding web services; e.g. the SOS database includes sensors and observations metadata that is compulsory for standards based implementations.

6.2. Service-based implementation

The OGC Sensor Web Enablement initiative defines an open-standards framework, upon which sensors data can be accessed, processed and shared by means of web-based implementations of specialized services aimed to provide specific capabilities based on certain necessities and feasibilities.

The requirements for this work, presented in Section 2.6 (Service-based platform requirements), delineate a platform design that includes three of these web services, as shown in Figure 4.1; this has led to the implementations described by Figure 6.1. Although, at least one alternative realization was made for each web service, in order to test the feasibility of replacing or updating the platform components.

In this regard, two implementations were carried out for the SOS component: 52N SOS and MS4W SOS (Section 5.2.2); even though, they both are fully compatible and able to work in parallel, the first one was selected because of the convenience of having the same provider as the component described below.

There were also considered three WPS implementations: INTAMAP, 52N WPS4R and pyWPS (Section 5.2.3); the first one was discarded because most of its capabilities are also available on WPS4R and also due to the fact that a straightforward setting up is only possible for Tomcat version 6 (see Section 5.1), while both 52N SOS and 52N WPS run on Tomcat version 8.

Despite the fact that pyWPS is also implemented and it works properly even in parallel with the other two services it is not being used for the present prototype, as its special capabilities are a desirable but not compulsory features by now, i.e. running Python based processes is not yet required and running R scripts is already covered by WPS4R; although, it can be useful for future web processes implementations.

Web services depend on a series of complementary resources such as the operating system's environmental variables or a web server and servlet container like Apache Tomcat as the repository to deploy the web applications (Section 5.2), among others.

However, each of these resources are commonly being developed by different organizations and not necessarily in parallel time periods; thus, very often the last released versions of a resource do not comply with the assumptions made for certain web service implementation.

For instance, INTAMAP was considered the ideal solution for the WPS component in the first place, because it has some particular features that are required, e.g. it is service-based and provides automatic interpolations. Nonetheless, due to Tomcat versions compatibility issues faced during the implementation it was impossible to configure it on the same operating system as the 52N SOS, which is also a core component of this design then INTAMAP WPS was then implemented on a different platform; thus, causing additional implementation time and costs plus an increase on the network traffic rate.

In general, web service implementations that are built-on different versions might have compatibility problems and even though they are fully OGC SWE standards compliant, they cannot cohabit without further efforts that imply high expertise level and additional costs as explained above.

Therefore, it is not completely possible to have a single and seamless platform due to the common mismatch between the mid-term implementations development and the short-term technologies involved. In a like manner, data models cannot be completely unified to form a single format to represent all the sensor data, because of the wide variety of phenomena described by them.

It is also relevant to mention that this study is limited to establish whether a component can be replaced or not in such a way that the whole platform still works properly, and neither efficiency indexes nor performance tests were included.

Therefore, this platform's interoperability lies on having an operational framework with all of the above described components based on standard specifications for the service and client implementations as well as the communication rules to facilitate interactions and messaging between them (e.g. requests and responses). In such a way that data integrity and consistency can be guaranteed by ensuring the utilization of standards compliant data models and formats to represent each's stage inputs and outputs.

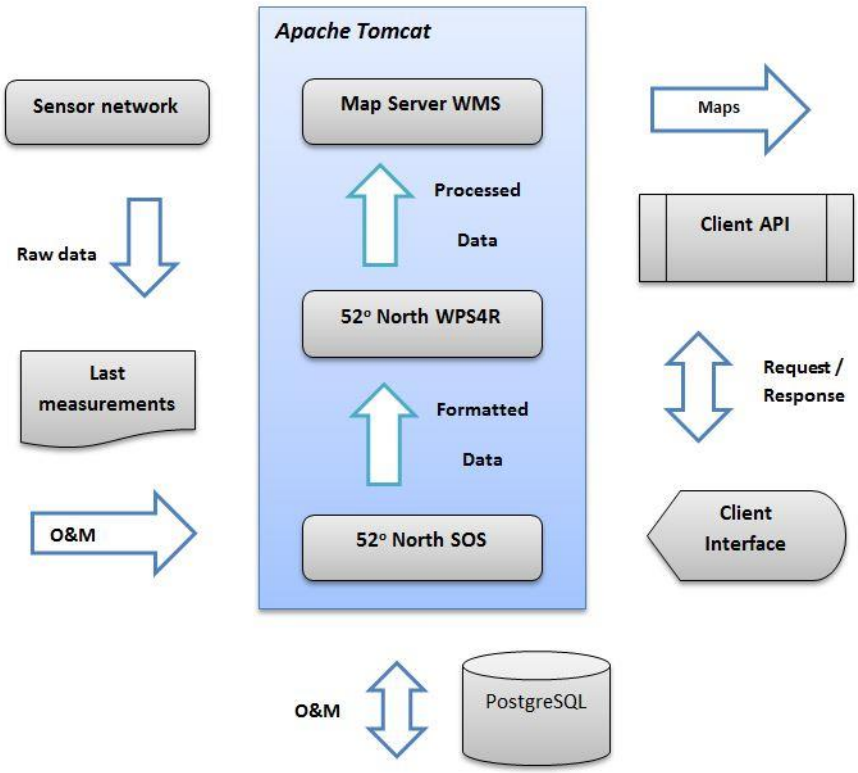


Figure 6.1 Implemented prototype's architecture

6.3. Geostatistical processing functions

All the elements described in Section 4.5 (Geostatistical functions) were implemented by using functions from special R libraries as described in Section 5.3.3, including: the histogram used to check the input data distribution, the special functions to automatically perform the interpolations and the cross-validation to assess the predictions accuracy.

The ordinary kriging method (Section 4.5.2) was selected for this study due to the fact that it is a widely used method for environmental variables, such as air pollution; and the interpolation results as well as the corresponding variance (or standard error) can be obtained through this method (Section 4.5.3); and also due to the fact that there is no other measurement available in this study's dataset that can be used as co-variable.

The variogram and the geostatistical interpolations were automatically generated through special functions that belong to the 'automat' and 'intamap' R packages as explained in Section 5.2.4 Geostatistical functions) and whose implementations are described in Section 5.3.3.

Furthermore, the method used by 'automap' to generate the variogram and the fitted variogram model is simpler than the one used by 'intamap'; in fact, 'intamap' selects a method from a list of options (which includes 'automap') based on the value set for the parameter 'maximumTime'.

An important limitation of nearly real-time interpolations is the fact that the empirical variogram has to be generated automatically by using pre-defined algorithms and the results could be too idealistic, e.g. the prediction's variability underestimations described in Section 5.3.3 (Validation); whereas a manually generated empirical variogram allows testing several models and values for the variogram parameters, hence more realistic results may be obtained by this means.

The accuracy assessment was performed by means of the diagnostic measurements derived from the cross-validation method as explained in Section 2.4.4 (Accuracy assessment); and these measurements were compared, as shown in Table 5.2. The obtained results show that 'intamap' predictions are slightly better in terms of accuracy than the ones obtained from 'automap' (Section 5.3.3 - Validation).

7. CONCLUSIONS AND RECOMMENDATIONS

The main objective of this research was to determine the feasibility of designing and implementing an interoperable service-based platform for sharing and automatically performing geostatistical functions on sensor data.

Specifically, the dataset provided by the AiREAS sensor network was used to perform spatial predictions; then the resulting air quality map and its corresponding uncertainty map are distributed through web-based interfaces in nearly real-time.

7.1. Conclusions

This main objective has been divided into three parts for this study (see Section 1.3), in order to achieve these; the research questions addressed in Section 1.4 are answered as follows:

Related to the first objective:

1. To design and implement an interoperable and standards-based platform for sharing and geostatistical processing air quality data.

1.1 What are the most suitable data models to share and process air quality data through web services?
As the input and output data specifications for OGC standards compliant services are built on XML schemas, such as O&M, UncertML, among others (Section 4.3); XML-based data models are most suitable for sharing and processing data throughout this platform.
These data models may be also used for the final stage of sharing, i.e. the output delivery to the client side. Although, it was established in Section 4.6 that JSON is most suitable for this stage, because it is a more concise data format, with a simpler structure; hence, no complex parsing is necessary on the client side.

1.2 Which architecture is appropriate to share properly the performed spatial predictions?
The system architecture presented in Figure 6.1 is based on the Service Oriented Architecture (SOA) approach (Section 4.6), as the main components are implemented through web services and the communication is being facilitated by standard data formats and protocols.
This architecture is considered appropriate because it provides a platform-independent way for accessing the data, as well as the for sharing the spatial interpolation results through web-based client interfaces, while preserving the data consistency and integrity by following the standards and specifications as described in Section 2.5.1.
Moreover, the proposed architecture has a mixed coarse-fine granularity with the web services (SOS, WPS and WMS) as the coarse grained blocks; and the pre-processing tasks plus the different client interfaces (Section 4.6) as the fine granular blocks.

1.3 What are the rules to ensure interoperability among the different platforms involved in this process?
OGC SWE standards, specifications and 'best practice' documentation for data models and web service implementations guarantee interoperability of the different components (Sections 2.5.1

and 4.3). Although, they do not cover the data pre-processing stage; and as the JSON format is not fully standardized they still rely on XML-based data models.

Related to the second objective:

2. To determine how spatiotemporal functions in general and spatial prediction in particular can be used to perform geostatistical processing on air quality data retrieved from sensor networks.

2.1 Which functions are most suitable to perform spatial prediction on air quality data retrieved from sensor networks?
Geostatistical interpolation methods like kriging are widely used to derive predictions on environmental sensor data (Section 2.4.2); besides these methods can be implemented on Web Processing Services either through 'Intamap' WPS or WPS4R (Section 5.3.3) as they allow the use of R scripts as web processes.

For this particular case, the ordinary kriging interpolation method was selected because as mentioned in Section 4.5.2 (Ordinary kriging) there is a spatial dependence which can be modelled from the input data (the variogram), and also because the utilized dataset does not provide another measurement that can be used as co-variable (Section 3.2-Data description).

2.2 How can the uncertainty of the spatial prediction results be determined?
Another reason to consider the kriging methods suitable for this study is the fact that they deliver not only the spatial predictions but also their uncertainty as explained in Section 4.5.3 (Accuracy assessment).

The kriging variance is a measure of how disperse are the results from the mean and it is computed for each prediction; thus the average of all the variances is used to determine the overall uncertainty and it was implemented on the present service-based platform through the 'krige.cv' function.

2.3 Which functions are the most appropriate to assess the outcome's quality?
The aforementioned diagnostic measurements (Table 5.2) derived from the cross-validation results are the RMSE as the overall precision indicator and MSDR as the ratio of the cross-validation residuals to the prediction errors (Section 5.3.3-Validation). The ME (the bias indicator) is not included because it is insensitive to inaccuracies in the variogram (Section 2.4.4).

In addition to the above described functions, the differences between 'intamap' and 'automap' cross-validation residuals were computed in order to compare the quality of the two implementations' outcome.

Related to third objective:

3. To provide a set of standardized functions that can be executed on web services, to automatically perform spatial prediction and to share the outcome with the client side.

3.1 Can web services be effectively used to receive data from a sensor network and to share the outcome through client applications?
Once the sensor data are properly pre-processed, when necessary, they are apt to be used as the input dataset; then an SOS can provide the means for retrieving and distributing these data (see

Section 4.3), and the WPS implementations considered in this work and described in Section 4.4 (Web Processing Service) are suitable for setting and executing R scripts as web processes.

During the present prototype development two SOS and three WPS implementations were realized; and despite the fact that 'Intamap WPS' is running on a different hardware platform, for the reasons explained in Section 6.2 it is working properly and can be used as well.

Nevertheless, compatibility issues have to be faced often when working with some components' versions developed by different organizations or launched in different time periods as mentioned in Section 6.2 (Discussion on service-based implementations)

3.2 How should the outcome be communicated to the different user types?
The special elements required by the 'advanced users' (Sections 3.3.2 and 3.4) to perform spatial predictions and to analyse the outcome, as described in Section 4.5 (Geostatistical functions), have to be provided on a web interface, as explained in Section 4.6 (Client side).

These elements are: a histogram displayed to allow checking the data distribution; a selection tool to allow choosing which interpolation function to apply; a plot of the generated variogram to check the spatial correlation; as well as the map displaying the performed predictions outcome and the corresponding uncertainty map. For the present prototype, as described in Section 5.3.2 this was achieved by using the web application framework called Shiny which provides all the above mentioned features on a web interface.

For the group of users defined as 'Developers' described in Section 3.4.3, as they need to know the platform architecture to adapt it to certain scenario, or to either replace or update components; they need the technical specifications described by technical means and tools such as the system programmer's manuals and UML elements like entity-relationship, class, state, use-case, and sequence diagrams.

3.3 What is the extent to which this process can be automated?
The web services and the client interface included in the described and implemented architecture (See Figures 4.1 and 6.1) are all interoperable as they all are standards-based and their inputs and outputs are being provided in standards compliant data models as described in Sections 4.3 (SOS), 4.4 (WPS) and 4.6 (Client side).

Hence, this interoperable collection of web implementations may be used as the underlying platform through which the involved components (sensor network, server and client) can automatically share and process sensor data. And, any of these implementations may be replaced or updated provided that the new component is also standards-based and its inputs and outputs are or can be transformed into the aforementioned standards compliant data models.

Furthermore, there are three stages within the geostatistical processing component that are critical for the process full automation: the pre-processing whose automation, as proposed in Section 5.2 (Technical setup), is addressed by coding appropriate scripts for each task (Section 4.2); the experimental variogram formulation and the variogram fitting, which can be tackled by using special functions like those provided by 'automap' and 'intamap' libraries for R and described in Section 5.3.3.

7.2. Recommendations

As explained in the introductory part of Chapter 5, due to time limitations only the first 4 stages of the present work were finished; hence further work is recommended on this project to implement the two missing components, i.e. the rest of the user interfaces and the user feedback component.

Since, it was one of this work's aims to provide nearly real-time interpolations from the provided sensor data, the input dataset utilized includes only the last ten-minute's measurements and due to the fact that exclusively the 'Airboxes' located within the city were used; after the filtering stage there are only thirty three sampling locations. This factor causes the automatically generated variogram to be rather vague and the interpolation variance to remain high; then further research is recommended by using spatiotemporal aggregation in order to improve the input dataset quality.

'Intamap' WPS and WPS4R are two WPS implementations that allow to build web services based on R scripts. Although, they both allow the R scripts deployment as web services, so it would be interesting to implement the 'intamap"s predefined R functions on WPS4R, to test the full compatibility of them.

Several OGC SWE web-service implementations have been realized during this study; nonetheless, no deep analyses were done on them in different scenarios (such as other operating systems and so on) and their corresponding APIs and web client platforms were not analysed at all; thus, such activities are recommended for further research projects.

LIST OF REFERENCES

52°North-Initiative. (2015a). Home - 52°North Initiative for Geospatial Open Source Software GmbH. Retrieved February 03, 2015, from http://52north.org/

52°North-Initiative. (2015b). WPS4R < Geostatistics < TWiki. Retrieved February 04, 2015, from https://wiki.52north.org/bin/view/Geostatistics/WPS4R

Aberer, K., Hauswirth, M., & Salehi, A. (2007). Infrastructure for Data Processing in Large-Scale Interconnected Sensor Networks. In *2007 International Conference on Mobile Data Management* (pp. 198–205). Conference and Custom Publishing.

AiREAS. (2014). AiREAS Eindhoven. Retrieved August 22, 2014, from http://eindhoven.aireas.com/

Akyildiz, I., Melodia, T., & Chowdury, K. (2007). Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, *14*(6), 32–39.

Alouani, A. T., & Rice, T. R. (1998). On optimal synchronous and asynchronous track fusion. *Optical Engineering*, *37*(2), 427.

Apache-Software-Foundation. (2015). Apache Tomcat - Welcome! Retrieved February 03, 2015, from http://tomcat.apache.org/

Atzeni, P., Bugiotti, F., & Rossi, L. (2014). Uniform access to NoSQL systems. *Information Systems*, *43*, 117–133.

Bachmaier, M., & Backes, M. (2008). Variogram or semivariogram? Understanding the variances in a variogram. *Precision Agriculture*, *9*(3), 173–175.

Bayraktar, H., & Turalioglu, F. S. (2005). A Kriging-based approach for locating a sampling site—in the assessment of air quality. *Stochastic Environmental Research and Risk Assessment*, *19*(4), 301–305.

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2008). OGC sensor web enablement: Overview and high level architecture. In *GeoSensor networks* (pp. 175–190). Springer.

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., … Winer, D. (2000). Simple object access protocol (SOAP) 1.1.

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., … Lemmens, R. (2011). New generation Sensor Web Enablement. *Sensors (Basel, Switzerland)*, *11*(3), 2652–99.

Burrough, P. A. (2001). GIS and geostatistics: Essential partners for spatial analysis. *Environmental and Ecological Statistics*, *8*(4), 361–377.

Chilès, J.-P., & Delfiner, P. (2009). *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons.

Chu, X., & Buyya, R. (2007). Service oriented sensor web. In *Sensor networks and configuration* (pp. 51–74). Springer.

Coburn, T. C., Yarus, J. M., Chambers, R. L., & others. (2005). *Stochastic Modeling and Geostatistics: Principles, Methods, and Case Studies, Vol. II, AAPG Computer Applications in Geology 5* (Vol. 5). AAPG.

Cressie, N. (1988). Spatial prediction and ordinary kriging. *Mathematical Geology*, *20*(4), 405–421.

Cressie, N. A. C., & Cassie, N. A. (1993). *Statistics for spatial data* (Vol. 900). Wiley New York.

Curran, P. J., & Atkinson, P. M. (1998). Geostatistics and remote sensing. *Progress in Physical Geography*, *22*(1), 61–78.

Da Cruz, P. S., Horne, R. N., & Deutsch, C. V. (2013). The Quality Map: A Tool for Reservoir Uncertainty Quantification and Decision Making. *SPE Reservoir Evaluation & Engineering*, *7*(01), 6–14.

DBU. (2009). Welcome to PyWPS — PyWPS. Retrieved February 04, 2015, from http://pywps.wald.intevation.org/

De Jesus, J., Barillec, R., Dubois, G., & Cornford, D. (2009, June 1). A web processing service for validating interpolation. *StatGIS 2009*.

Diao, Y., Ganesan, D., Mathur, G., & Shenoy, P. J. (2007). Rethinking Data Management for Storage-centric Sensor Networks. In *Conference on Innovative Data Systems Research* (pp. 22–31).

EC. (2009). INTeroperability and Automated MAPping: INTAMAP. Retrieved February 04, 2015, from http://www.intamap.org/

Foerster, T., Nüst, D., Bröring, A., & Jirka, S. (2012). Discovering the Sensor Web through Mobile Applications. In G. Gartner & F. Ortag (Eds.), *Advances in Location-Based Services* (pp. 211–224). Springer Berlin Heidelberg.

Franke, R. (1982). Scattered data interpolation: tests of some methods. *Mathematics of Computation*, *38*(157), 181–181.

Ganesan, D., Estrin, D., & Heidemann, J. (2003). Dimensions. *ACM SIGCOMM Computer Communication Review*, *33*(1), 143–148.

Ganesan, D., Greenstein, B., Perelyubskiy, D., Estrin, D., & Heidemann, J. (2003). An evaluation of multi-resolution storage for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03* (p. 89). New York, New York, USA: ACM Press.

Gao, H., Wang, J., & Zhao, P. (1996). The updated kriging variance and optimal sample design. *Mathematical Geology*, *28*(3), 295–313.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation* (p. 483). Oxford University Press.

Granell, C., Diaz, L., & Gould, M. (2007). Managing Earth observation data with distributed geoprocessing services. In *2007 IEEE International Geoscience and Remote Sensing Symposium* (pp. 4777–4780).

Granell, C., Gould, M., Manso, M. Á., & Bernabé, M. Á. (2009). *Handbook of Research on Geoinformatics*. (H. A. Karimi, Ed.). IGI Global.

Havlik, D., Bleier, T., & Schimak, G. (2009). Sharing Sensor Data with SensorSA and Cascading Sensor Observation Service. *Sensors (Basel, Switzerland)*, *9*(7), 5493–5502.

HDF-Group. (2014). HDF Group - HDF5. Retrieved January 06, 2015, from http://www.hdfgroup.org/HDF5/

Ho Lee, S., Hoon Han, J., Taik Leem, Y., & Yigitcanlar, T. (2008, July 21). Towards ubiquitous city : concept, planning, and experiences in the Republic of Korea. *Knowledge-Based Urban Development : Planning and Applications in the Information Era*. IGI Global, Information Science Reference.

Huang, C.-F., & Tseng, Y.-C. (2005). The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, *10*(4), 519–528.

INSPIRE-Directive. (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Published in the Official Journal on the 25th April*.

Institute for Statistics and Mathematics. (2015). what is R. Retrieved from http://www.r-project.org/

Jindal, A., & Psounis, K. (2004). Modeling spatially-correlated sensor network data. In *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.* (pp. 162–171). IEEE.

Juba, S. Samoladas, V. Boretos, N. Manakos, I. Karydas, C. G. (2007). IMS: a Web-based Map Server for Spatial Decision Support. *Neural, Parallel & Scientific Computations.*, *15*(2), 207–220.

Kumar, S. P. (2003). Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, *91*(8), 1247–1256.

Kyriakidis, P. C., & Journel, A. G. (1999). Geostatistical Space–Time Models: A Review. *Mathematical Geology*, *31*(6), 651–684.

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, *36*(6), 47–56.

Lee, K. K.-Y., Tang, W.-C., & Choi, K.-S. (2013). Alternatives to relational database: comparison of NoSQL and XML approaches for clinical data storage. *Computer Methods and Programs in Biomedicine*, *110*(1), 99–109.

Li, J., & Heap, A. D. (2008). *A review of spatial interpolation methods for environmental scientists* (2008/23 ed., Vol. 137, p. 137 pp). Canberra, Australia: Geoscience Australia Canberra.

Liu, L., & Özsu, M. T. (2008). Encyclopedia of database systems. *Liu & T. Ozsu, Eds. Encyclopedia of Database Systems*.

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

Moniruzzaman, A. B. M., & Hossain, S. A. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison.

Na, A., Priest, M., Niedzwiadek, H., & Davidson, J. (2007). OGC Implementation Specification 06-009r6: Sensor Observation Service.

Nance, C., Losser, T., Iype, R., & Harmon, G. (2013). NOSQL VS RDBMS - Why there is room for both. *SAIS 2013Proceedings*.

Nittel, S., Labrinidis, A., & Stefanidis, A. (Eds.). (2008). *GeoSensor Networks* (Vol. 4540). Berlin, Heidelberg: Springer Berlin Heidelberg.

Nüst, D., Stasch, C., & Pebesma, E. (2011). Connecting R to the sensor web in Geertman, S. In F. Reinhardt, W. & Toppen (Ed.), *Advancing Geoinformation Science for a Changing World* (1st ed., pp. 227 – 246). Springer Berlin Heidelberg.

OGC. (2015a). Open Geospatial Consortium | OGC. Retrieved January 06, 2015, from http://www.opengeospatial.org/

OGC. (2015b). Sensor Observation Service (SOS) | OGC. Retrieved January 11, 2015, from http://www.opengeospatial.org/standards/sos

OGC. (2015c). Sensor Web Enablement (SWE) | OGC. Retrieved January 11, 2015, from http://www.opengeospatial.org/ogc/markets-technologies/swe

OGC. (2015d). Web Processing Service | OGC. Retrieved January 14, 2015, from http://www.opengeospatial.org/standards/wps

Pebesma, E., Cornford, D., Dubois, G., Heuvelink, G. B. M., Hristopulos, D., & Pilz, J. et al. (2011). INTAMAP: The design and implementation of an interoperable automated interpolation web service. *Computers & Geosciences*, *37*(3), 343–352.

Perumal Murugan, A. S. (2013). A Study of NoSQL and NewSQL databases for data aggregation on Big Data. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:706302

Poslad, S. (2009). *Ubiquitous Computing*. Chichester, UK: John Wiley & Sons, Ltd.

Python Software Foundation. (2015). About Python™ | Python.org. Retrieved January 22, 2015, from https://www.python.org/about/

Řezník, T. (2007). Sensor Web Enablement. Retrieved from http://www.muni.cz/research/publications/725884

Robinson, T. P., & Metternicht, G. (2006). Testing the performance of spatial interpolation techniques for mapping soil properties. *Computers and Electronics in Agriculture*, *50*(2), 97–108.

Rodriguez, A. (2008). Restful web services: The basics. *IBM developerWorks*.

Rossi, R. E., Dungan, J. L., & Beck, L. R. (1994). Kriging in the shadows: Geostatistical interpolation for remote sensing. *Remote Sensing of Environment*, *49*(1), 32–40.

Rossiter, D. G. (2012). Co-kriging in gstat - R_ck.pdf. *Techical note*.

RStudio. (2015). Shiny. Retrieved February 05, 2015, from http://shiny.rstudio.com/

Schmidt, J. W., Ceri, S., & Missikoff, M. (Eds.). (1988). *Advances in Database Technology—EDBT '88* (Vol. 303). Berlin, Heidelberg: Springer Berlin Heidelberg.

Scholes, R. J., Mace, G. M., Turner, W., Geller, G. N., Jürgens, N., Larigauderie, A., … Mooney, H. A. (2008). Toward a global biodiversity observing system. *Science*, *321*(5892), 1044–1045.

Schut, P., & Whiteside, A. (2007). OpenGIS Web processing service. *OpenGIS Standard, Version 1.0. 0, OGC 05-007r7*.

Singh, V., Carnevale, C., Finzi, G., Pisoni, E., & Volta, M. (2011). A cokriging based approach to reconstruct air pollution maps, processing measurement station concentrations and deterministic model simulations. *Environmental Modelling & Software*, *26*(6), 778–786.

Strang, T., & Linnhoff-Popien, C. (2004, September 7). A Context Modeling Survey. *Workshop Proceedings*.

The PostgreSQL Global Development Group. (2015). PostgreSQL: About. Retrieved January 22, 2015, from http://www.postgresql.org/about/

Tilak, S., Abu-Ghazaleh, N. B., & Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, *6*(2), 28–36.

Tobler, W. (2004). On the First Law of Geography: A Reply. *Annals of the Association of American Geographers*, *94*(2), 304–310.

Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, *46*, pp. 234–240.

Tudorica, B. G., & Bucur, C. (2011). A comparison between several NoSQL databases with comments and notes. In *2011 RoEduNet International Conference 10th Edition: Networking in Education and Research* (pp. 1–5). IEEE.

UCAR. (2015). NetCDF. Retrieved February 01, 2015, from https://www.unidata.ucar.edu/software/netcdf/docs/

Usländer, T. (2005). RM-OA-Reference Model for the ORCHESTRA Architecture. *Deliverable D3*, *2*, 5–107.

Usländer, T., & others. (2009). *Specification of the sensor service architecture, Version 3.0 (Rev. 3.1)*.

Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, *52*(1), 40.

Wackernagel, H. (2003). *Multivariate geostatistics*. Springer.

Wang, Y.-C., Hsieh, Y.-Y., & Tseng, Y.-C. (2008). Compression and Storage Schemes in a Sensor Network with Spatial and Temporal Coding Techniques. In *VTC Spring 2008 - IEEE Vehicular Technology Conference* (pp. 148–152). IEEE.

Webster, R., & Oliver, M. A. (2007). *Geostatistics for Environmental Scientists (Google eBook)* (p. 330). John Wiley & Sons.

Weiser, M. (1993). Hot topics-ubiquitous computing. *Computer*, *26*(10), 71–72. Retrieved from http://www.computer.org/csdl/mags/co/1993/10/rx071-abs.html

WHO. (2014). WHO | Ambient (outdoor) air quality and health. World Health Organization. Retrieved from http://www.who.int/mediacentre/factsheets/fs313/en/