

# Web-based geostatistics using WPS

Jorge de Jesus<sup>1</sup>, Grégoire Dubois<sup>2</sup> Paul Hiemstra<sup>3</sup>

<sup>1</sup>Institute for Environment and Sustainability / Joint Research Centre of the European Commission  
jorge.de-jesus@jrc.it

<sup>2</sup>Institute for Environment and Sustainability / Joint Research Centre of the European Commission  
gregoire.dubois@jrc.it

<sup>3</sup>Department of Physical Geography / University of Utrecht  
p.hiemstra@geo.uu.nl

**Abstract.** Mapping data using geostatistics can be a time-consuming process because of the many parameters to define. `automap`, a geostatistical package written in R, was developed to define automatically a spatial correlation model, a step that is considered to be the biggest obstacle for automating the spatial interpolation process with geostatistical algorithms. The implementation of `automap` into a Service Oriented Architecture (SOA) would ensure the automatic interpolation routines to be platform independent, easy to integrate into existing applications, and freely available in a network environment. `automap` is here further proposed as a Web Processing Service (WPS) to allow end-users to benefit from a remote service to be used for the real-time mapping of environmental data in routine situations. It is the purpose of this paper to present the architecture of the service as well as to discuss by means of case studies the limitations of the real time mapping system for emergency response.

## 1 INTRODUCTION

Geostatistics is nowadays commonly used to map environmental variables for risk management. Information about the quality of the estimated values can be easily generated and, in contrast to other spatial interpolation functions, geostatistics takes a model of the spatial process into account (see e.g. Goovaerts 1997, Chiles and Delfiner 1999). The well-defined theoretical framework of the methods and the possibility to produce information on the associated uncertainty are likely to be the two main reasons for applying geostatistics to support decision making problems. Progress made in geoinformatics also encouraged the use of geostatistics by providing the scientific community with the tools required for the exploratory spatial data analysis (ESDA). ESDA does not only ease the detection of anomalies but

it facilitates also the detection and the modelling of the spatial correlation of the analysed process. The drawback in providing users with advanced tools for the ESDA as well as with the large number of estimators falling under the generic name of kriging is that the interpolation process turns out to be more and more complex (Jarvis, Stuart and Cooper 2003). Consequently, using geostatistics for real-time applications is becoming cumbersome.

This difficulty in developing automatic interpolation system has been recently highlighted through an exercise (EUR 2005) in which participants were invited to produce maps in near real-time. One will therefore not be surprised that, although many environmental monitoring networks do collect in near real-time data such as air quality measurements, radioactivity levels, temperatures or seismic observations, only a very few networks are linked to a system generating maps automatically from the collected data (Brenning and Dubois 2008). It is the purpose of this paper to present some recent work done towards the development of services providing means for interpolating data automatically as well as to remind the readers about the main risks in providing information without any human intervention.

## 2 AUTOMATIC INTERPOLATION OF A DATASET

### 2.1 automap

A few advanced statistical algorithms for the automatic interpolation of spatial data have been developed in the frame of the Spatial Interpolation Comparison<sup>1</sup> 2004 exercise (EUR 2005) but the boundaries of their condition of use remain to be largely explored. Among these tools, `automap` 0.4-1 (Hiemstra et al. 2008) was developed as an open source algorithm depending on `gstat` 2.5.1 (Pebesma 2004) and was also written with the statistical programming language R (R Development Core Team 2008). `automap` was initially developed to define automatically a spatial correlation model, the so-called “variogram”, a step considered to be the biggest obstacle for the automation of any interpolation process based on geostatistics.

### 2.2 Kriging

Various estimators with different applications and scopes exist in geostatistics. In `automap`, the estimator that is used is the one that is most commonly encountered, namely Ordinary Kriging (OK). OK is also known as

---

<sup>1</sup> <http://www.ai-geostats.org/index.php?id=events>

the Best Unbiased Linear Estimator (BLUE) as it minimizes the squared prediction error, is a weighted linear combinations of the available data and unbiased because the mean of the prediction error is zero. OK relies on the variogram, a model of the spatial correlation structure of the data, to determine the weighting values. We refer to the standard literature (see e.g. Cressie, 1993; Goovaerts, 1997; Chiles and Delfiner, 1999) for more information about the theoretical background.

### 2.3 Variography

Fitting a model of the spatial correlation (step 2 above) is known to be the most critical step in geostatistics.

On the basis of the observations, one first has to calculate the sample variogram for distance intervals (bins)  $(h_k, h_k + \delta_k)$  using:

$$\hat{\gamma}(\bar{h}_k) = \frac{1}{2N_k} \sum_{i=1}^{N_k} [\hat{e}(x_i) - \hat{e}(x_i + h)]^2, \quad \forall (x_i, x_i + h) : h \in [h_k, h_k + \delta_k] \quad (2)$$

where  $\bar{h}_k$  is the average semivariance for bin  $k$ ,  $\hat{e}(x_i)$  and  $\hat{e}(x_i + h)$  the estimated residuals at location  $x_i$  and  $x_i + h$ ,  $N_k$  the number of pairs of points for bin  $k$ .

An important constraint when fitting a model on the sample variogram is that the model has to be chosen among functions that are *conditionally positive definite* (Chiles and Delfiner 1999).

### 2.4 Generating the output results

Once a variogram model is chosen and fitted, the predictions and the associated kriging variances can be calculated for each defined location, usually a number of nodes defined on a regular grid. Assuming that the kriging variance is normally distributed around the kriging prediction, one can construct approximate 95% prediction intervals with

$$\left[ \hat{Z}(x_0) - 1.96\sigma(x_0), \hat{Z}(x_0) + 1.96\sigma(x_0) \right] \quad (3)$$

where  $\sigma(x_0)$  is the square root of the kriging variance.

The prediction interval can so be used to assess whether a certain threshold has been exceeded by our variable or not with 95% certainty. This type of information is obviously useful for decision-makers when judging on the needs to take counter-measures.

## 2.5 Automatic Kriging

In the case of an environmental emergency (heavy smog, earthquake, release of radioactivity in the environment), one would expect from decision-makers to quickly and accurately delineate the risk zones, something that would be greatly facilitated by minimizing the time needed to analyse the data. By choosing and fitting automatically the variogram, `automap` does not only facilitate the time-consuming process of modelling the spatial correlation structure of the investigated variable but it also allows the batch processing of the data for further input in models. However, the variogram modelling is not the only obstacle to using geostatistics for the real-time mapping in situations of emergencies. Among other issues, the automatic identification of strong geometrical anisotropies, the automatic detection and interpretation of outliers or the approach to be followed when dealing with non-Gaussian distributions (see EUR 2003, EUR 2005 for discussions on automating geostatistics) are still largely open for debate.

## 3 CASE STUDIES WITH `automap`

To illustrate our purposes, we will apply `automap` to a set of daily rainfall measurements made in Switzerland (EUR 2003) as well as two sets of daily observations of gamma dose rates, one presenting a few simulated values illustrating a potential accidental release of radioactivity in the atmosphere (EUR 2005). These datasets have been used in two Spatial Interpolation Comparison (SIC) exercises organized in 1997 (SIC97) and 2004 (SIC2004), respectively. In these exercises, participants were invited to estimate values of a variable at given locations by using information provided by a subset of the total number of measurements of the variable. The algorithms of the participants were then contrasted with each other by comparing the estimation errors of each spatial interpolation technique. While in SIC97 participants had a few weeks time to submit their results, the algorithms used in SIC2004 were supposed to be fully automatic and the results had to be sent back within the shortest time.

### 3.1 Automatic fitting of the experimental variogram

While the participants to the SIC exercises had in both editions information about the network topology to allow them to design their algorithms, the approach adopted in `automap` is more generic. When calculating the experimental semivariogram, the bins are first divided over the range of distances for which the sample variogram is requested, placing more bins at shorter distances. An initial guess of the variogram model is used as a starting point for iteratively reweighted least squares fitting of the variogram

model (Gauss-Newton fitting, (Cressie 1993)), where the weights are equal to  $N_k / \bar{h}_k^2$ .

The nugget effect, the range and the sill are initially set in the following way.

- The nugget effect (the value of the semivariance for distances close to 0) can be set as the value taken by the smallest semivariance in the sample variogram.
- The range is defined as 0.35 times the diagonal length of the study area;
- The sill is computed as the average of the maximum and the median semivariances of the sample variogram.

Four functions have been implemented by default as potential candidates for the semivariogram model: the Gaussian, spherical, Matern and exponential models have been here considered. From these four models, the one that will lead to the smallest error (defined here by the sum of squares) with the sample variogram is chosen.

### **3.2 AUTOMAP applied to the SIC data**

Figures 1, 2 and 3 show the outcome of the results of `automap` when applied to the SIC data. With the automatically fitted variograms, these figures show the kriging predictions (upper left plots) and associated kriging variances (upper right plots) for the daily rainfall measurements used in SIC97 (Figure 1), the daily gamma dose rates in routine situation (Figure 2) or in case of anomalies/emergencies (Figure 3) as used in SIC2004. The interpolated grids produced in `automap` can easily be stored in different formats using either GML (Geographical Markup Language) or GeoTIFF.

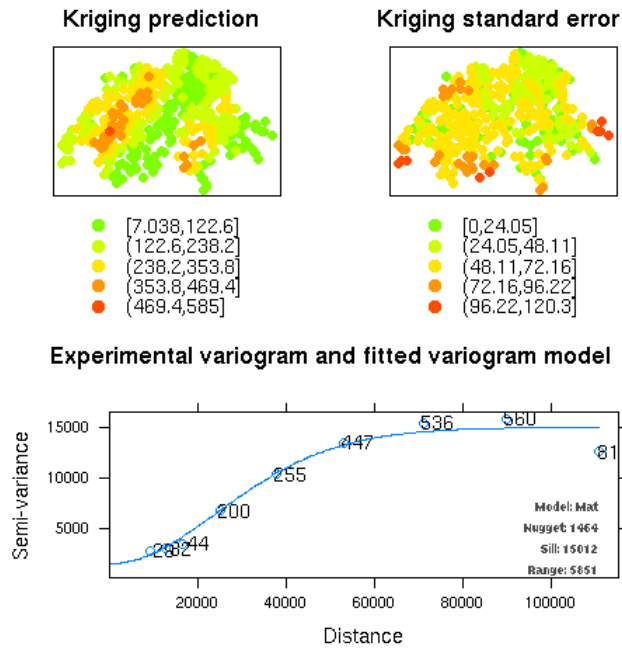


Figure 1: Kriging predictions (upper left) and associated kriging variances (upper right) for the daily rainfall measurements used in SIC97. The lower figure shows the automatically selected Matern variogram model.

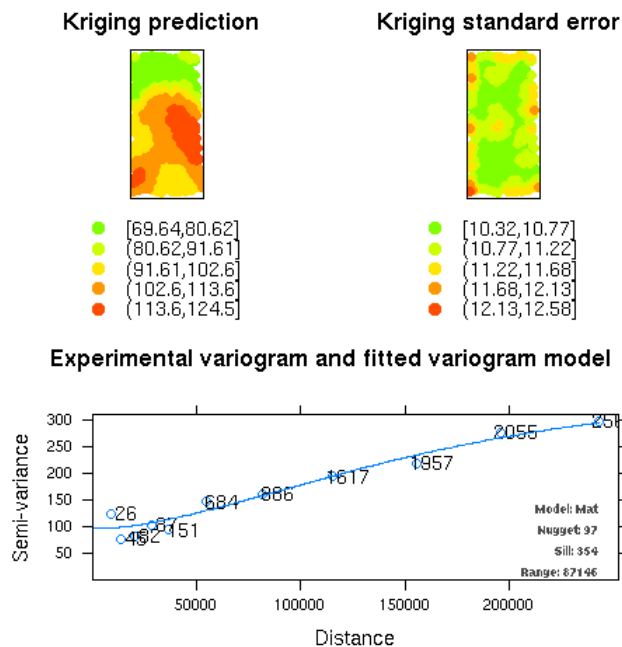


Figure 2: Kriging predictions (upper left) and associated kriging variances (upper right) for the gamma dose rates without anomalies used in SIC2004. The lower figure shows the automatically selected Matern variogram model.

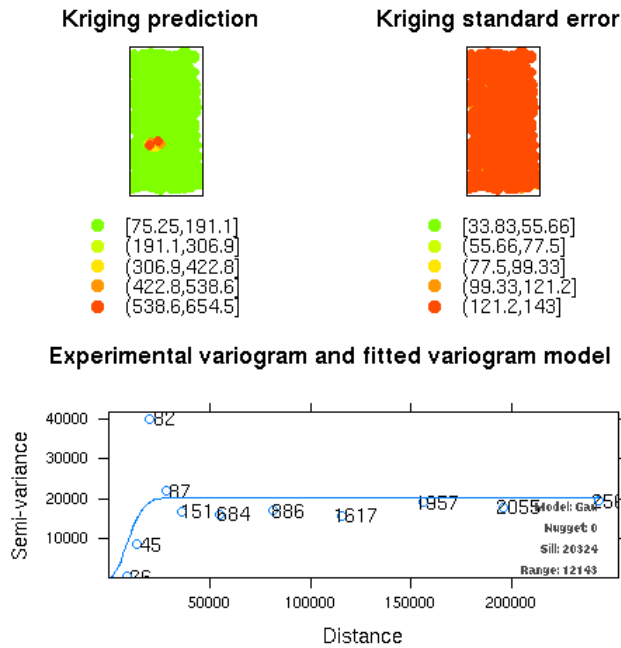


Figure 3: Kriging predictions (upper left) and associated kriging variances (upper right) for the gamma dose rates with anomalies used in SIC2004. The lower figure shows the automatically selected spherical variogram model.

### 3.3 Generating the outputs

The outputs of `automap` are set by default as a regular grid covering the area. The plots of the predicted values against the observations are shown in Figure 4. The summary statistics of the errors (Mean Absolute Errors, the Root Mean Squared Error and the regression coefficients of the plots associated with Figure 4) obtained with `automap` are shown in Table 1. This table can be compared with Table 2 that is giving the best results obtained from the participants of the exercises.

From these tables, one can draw a few conclusions. `automap` managed to generate excellent results for the two case studies that did not incorporate large anomalies, a situation that is typical for many environmental processes that can be considered as “normal” situations. Interestingly enough, `automap` generated the overall best results in the case of SIC97. This is most probably due to the use of the Matern variogram model that was not often used a decade ago and one would expect today, 10 years after the first SIC exercise, to have better results from the experts. Still, these results show that it is nowadays not unrealistic to imagine in the near future interpolation algorithms, based on geostatistics, used for the real-time mapping of environmental variables in situations of routine.

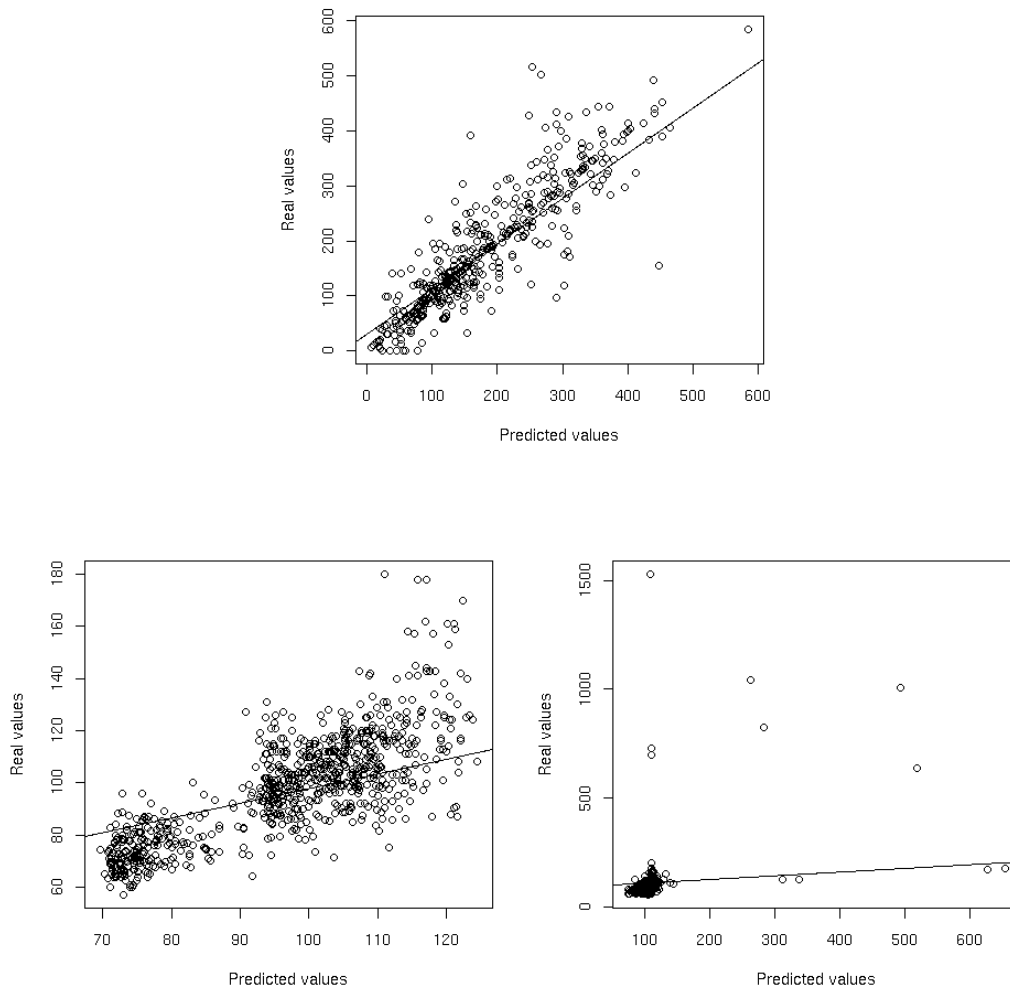


Figure 4: Plots of the predicted values against the true observations for the SIC97 (above) and SIC2004 (below) data. The lower left plot shows the gamma dose rates data without anomalies, the lower right plot shows the results for the data containing the anomalies

Table 1: Summary tables with SIC results obtained from automap. NA = Not Available.

<i>Results obtained from automap</i>	MAE	RMSE	<i>r</i>
SIC97: daily rainfall	31.8	51.4	NA
SIC2004: daily gamma dose rates	9.2	12.5	0.78
SIC2004: daily gamma dose rates + simulated values	23.2	76.9	0.40



Table 2: Summary tables with best results obtained from the participants to the SIC exercises

<i>Best results obtained from participants to SIC exercises</i>	MAE	RMSE	<i>r</i>
SIC97: daily rainfall	32.0	53.1	NA
SIC2004: daily gamma dose rates	9.1	12.4	0.79
SIC2004: daily gamma dose rates + simulated anomalies	14.9	45.46	0.86

Cases with anomalies are obviously much more complex. If `automap` generated results, in this special case, that are very comparable to the best results obtained from other geostatistical algorithms (not shown here), the errors obtained are still much higher than the best results indicated in Table 2 which were reported by a Support Vector Regression algorithm.

With more work required to understand the boundary conditions of the use of `automap` for the real-time mapping of environmental variables, more research is clearly needed for mapping data with extreme values that often characterize emergency situations. These issues are currently explored in the frame of the INTAMAP<sup>2</sup> project.

## 4 WEB PROCESSING SERVICE

### 4.1 Towards a web based version of `automap`

In addition to having a number of statistical problems to overcome, an automatic interpolation algorithm would also need to be easy to integrate into a monitoring system. This last issue can be solved by adopting a service oriented architecture (SOA) as this provides a powerful framework for developing an automatic interpolation service to be utilised by anyone capable of employing web services (Pebesma, Cornford and Dubois 2008). A SOA is a collection of services with communication abilities that are distributed over a network, allowing a web client to access and run a service and to get back some results from it (Izza, Vincent and Burlat 2008). An interpolation procedure can therefore be considered as a client-to-service interaction in which a user/client sends data to an interpolation service that will respond with interpolated data in the form of a map. Using this phi-

---

<sup>2</sup> INTAMAP stands for Interoperability and Automated Mapping. INTAMAP ([www.intamap.org](http://www.intamap.org)) is an international research project funded the DG INFSO of the European Commission

losophy a system measuring environmental parameters can request an interpolated map from its data to some service provider physically located in any place with Internet access.

Implementing automap into a SOA would ensure that routines become platform independent, easy to integrate into existing applications, and freely available in a network environment. Hence, proposing automap as a web service could provide end-users with a useful tool to produce maps in real-time for the monitoring of environmental variables in situations of routine or simply for batch processing of data.

### 4.2 Setting up Web Processing Services (WPS)

WPS is a web service following a Service Oriented Architecture (SOA) model where pre-programmed geospatial calculations and/or computation models are available as dispersed services/processes over a network. Note that version 1.0 of the OpenGIS® WPS Interface Standard was approved recently (February 2008) by the Open Geospatial Consortium (OGC). The SOA model allows processes to be available in a transparent way, independent of operating system and programming languages used for their creation.

WPS supports three operations: *GetCapabilities*, *DescribeProcess* and *Execute* (Figure 5).

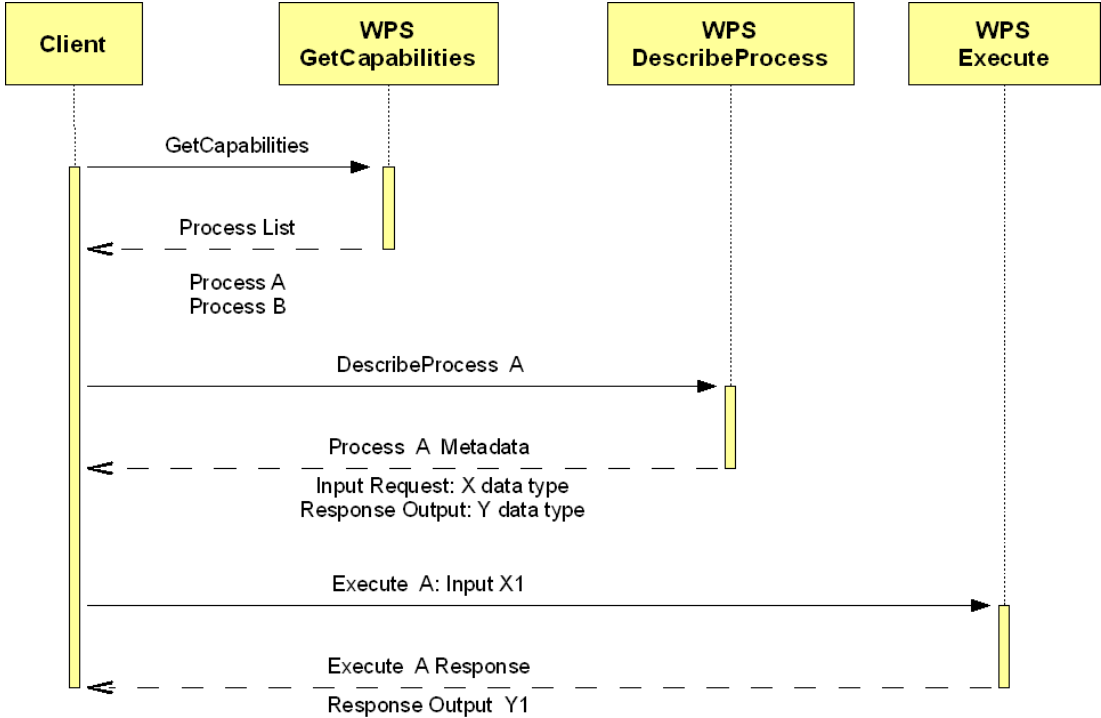


Figure 5: General procedure for WPS requests/response

*GetCapabilities* and *DescribeProcess* are descriptive operations that provide the necessary metadata to the *Execute* request. *GetCapabilities* allows for clients to request a document listing the processes served. A *DescribeProcess* operation indicating a specific process provides detailed information about this process as well as the input/output parameters and formats that are required for the proper coding of the *Execute* request corresponding to the desired process.

WPS supports two methods for service querying: Key-Value-Pairs/GET or XML file/POST. Each operation defines for a client the standard method to make a request. A *GetCapabilities* request is usually done using the HTTP-GET method using Key-Value-Pairs (KVP) to the server providing the services, as shown in the next example:

```
http://remwps.jrc.it/wps.py?request=GetCapabilities&service=WPS
```

The *Execute* operation can be requested using KVP, for simple operations with basic input/output but, for the majority of the geospatial processes, a request using a XML file using a HTTP-POST method needs to be made. The XML *Execute* request will embed the input data necessary for the process as shown in Table 3.

The server containing the process will accept the *Execute* request and respond with a XML document containing the output or the location of the output, as shown in Table 4.

XML responses and requests are defined in WPS and validated through schemas designed by OGC (2002, 2005).

### 4.3 System Architecture

RAISIN (REM<sup>3</sup>'s Automatic Interpolation Service for INTAMAP) is a web service providing as a WPS automatic interpolation using the R 2.6.1 codes of automap. With the general *GetCapabilities*, *DescribeProcess* and *Execute* requests defined by the WPS, RAISIN accepts a set of point data and sends back interpolated points for a specific grid.

---

<sup>3</sup> REM stands for the Radioactivity Environmental Monitoring group of the Institute for the Environment and Sustainability of the Joint Research Centre of the European Commission <http://rem.jrc.ec.eu.int/>

Table 3: Example of an XML/POST *Execute* request. The XML request indicates the process with the <ows:Identifier> tag and embedded input data as a GML structure inside the <ComplexValue> tag. This XML is only for illustration purposes since it only provides two data points.

```

<?xml version="1.0" encoding="UTF-8" ?>
<Execute service="WPS" version="0.4.0" store="false" status="false"
xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wps
http://www.opengeospatial.net/wps.\wpsExecute.xsd">
  <ows:Identifier>autokrigeprocess</ows:Identifier>
  <DataInputs>
    <Input>
      <ows:Identifier>XYZPoints</ows:Identifier>
      <ows:Title>XYZ SIC97 data points as GML</ows:Title>
      <ComplexValue schema="http://ogr.maptools.org/sic97.xsd">
<ogr:FeatureCollection
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ogr.maptools.org/sic97.xsd"
  xmlns:ogr="http://ogr.maptools.org/"
  xmlns:gml="http://www.opengis.net/gml">
  <gml:featureMember>
    <ogr:sic97 fid="F0">
      <ogr:x>181072</ogr:x>
      <ogr:y>33361</ogr:y>
      <ogr:z>3.009</ogr:z>
    </ogr:sic97>
  </gml:featureMember>
  <gml:featureMember>
    <ogr:sic97 fid="F1">
      <ogr:x>181025</ogr:x>
      <ogr:y>333558</ogr:y>
      <ogr:z>3.057</ogr:z>
    </ogr:sic97>
  </gml:featureMember>
</ogr:FeatureCollection>
</ComplexValue>
    </Input>
  </DataInputs>
  <!--DataOutput, like DataInputs are not mandatory-->
  <!-- No need to define DataOutput -->
</Execute>

```

Table 4: Result of the *Execute* request of Table 3. The response only contains the server location of the result of the process (in this case a GeoTIFF).

```

<?xml version="1.0" ?>
<ExecuteResponse version="0.4.0"
xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wps
http://www.ogcnetwork.net/schemas/wps/0.4.0/wpsExecute.xsd">
  <ows:Identifier>autokrigeprocess</ows:Identifier>
  <Status><ProcessSucceeded/></Status>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>GeoTiff</ows:Identifier>
      <ows:Title>Interpolated Output Map</ows:Title>
      <ows:Abstract>Resulting interpolated Map. With X,Y coordi-
nates, predicted value (var1.pred), standard-deviation (var1.stdevev) and variance
(var1.var), for each interpolated point, as a link to a stored GeoTIFF. The different re-
sults will be allocated to different bands: band1 == var1.pred, band2==var1.stdev and
band3==var1.var</ows:Abstract>
      <ComplexValueReference encoding="utf-8" format="image/tif"
ows:reference="http://remwps.jrc.it/wpstmp/interpolation176304.tif" schema=""/>
    </Output>
  </ProcessOutputs>
</ExecuteResponse>

```

RAISIN was built in a Linux-Apache-Python-R (LAPR) system using PyWPS as the major API for WPS requests and R for running the *auto-map* package containing the interpolation algorithms, following the SOA model with the objective of serving *automap*'s functions/functionalities as web processes.

Apache 2.2.4 is running under Linux to provide web service functionalities, mainly listening to network request. It also contains the `mod_python` 3.3.1 module that executes the python 2.5.1 script (`wps.py`) that provides the WPS implementation.

The script (`wps.py`) is the main core of RAISIN and is based on PyWPS 1.0.0 with `mod_python` support instead of the native CGI capabilities of PyWPS. Apache supports two methods for script execution: CGI or module. In the CGI case, the script is executed by launching a new system thread that will execute the script. In the `mod_python`, a module with the

python engine is loaded when Apache starts and will run the script internally. The `mod_python` option is more efficient but it requires a different method for passing information from Apache to the script.

The connection between the python script and R is made using the Rpy 1.0-RC3 module which allows R sessions and functions to be accessed within the Python script.

Following the general WPS procedure, a client will use the *GetCapabilities* and *DescribeProcess* requests to obtain the metadata of the process, executing later an interpolation request with *Execute* and sending the data back in the predefined format (Figure 6).

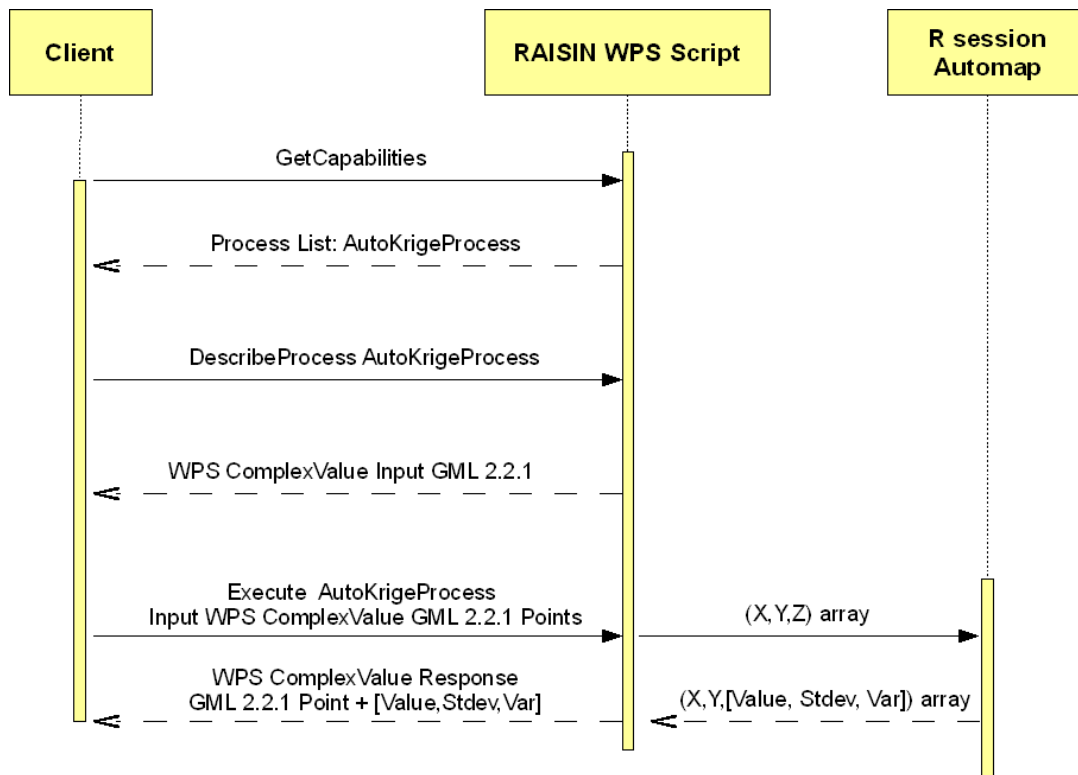


Figure 6: Processing of user requests by the RAISIN WPS server

#### 4.4 Input/output data

Input/Output requirements are described in the response XML of the *DescribeProcess* request for the *AutoKrigeProcess* (Table 5). In the *AutoKrigeProcess* the client submits data for interpolation as a GML version 2.1.2 (Open Geospatial Consortium Inc., 2002) structure where each point is represented as a feature member contained in one feature collection.

Table 5: *DescribeProcess* request and XML response.

http://remwps.jrc.it/wps.py?request=describeprocess&  
identifier=autokrigeprocess2&service=wps&version=0.4.0

```
<ProcessDescriptions      xsi:schemaLocation="http://www.opengeospatial.net/wps
http://www.ogcnetwork.net/schemas/wps/0.4.0/wpsDescribeProcess.xsd">
  <ProcessDescription processVersion="0.1" statusSupported="false"      store-
Supported="false">
    <ows:Identifier>autokrigeprocess</ows:Identifier>
    <ows:Title>AutoKrige Process of GeoTIFF</ows:Title>
    <ows:Abstract>WPS implementation of the AutoKrige function con-
tained on the automap package of R. The autokrige automates the kriging process mak-
ing it a simple input/output process with out the need for human intervention in semi
variogram building and model choosing. Output as Geotiff file</ows:Abstract>
  <DataInputs>
    <Input>
      <ows:Identifier>XYZPoints</ows:Identifier>
      <ows:Title>XYZ points of know points in the domain</ows:Title>
      <ows:Abstract>Input point for AutoKrige function. Input points are in GML
version 2.1.2 format obtained from the conversion of a simple CSV (containing XYZ)
using OGR from GDAL 1.4.2.0 (ogr2ogr <output file .gml> <input file .csv> -f
"GML"). The GML structure will be embedded in the Input of the XML request as a
complex value THEREFORE, EXECUTE IS REQUESTED USING XML AND NOT
KVP. Undefined namespaces will produce a bad XML exception</ows:Abstract>
      <ComplexData defaultFormat="text/xml">
        <SupportedComplexData>
          <Format>text/xml</Format>
        </SupportedComplexData>
      </ComplexData>
      <MinimumOccurs>1</MinimumOccurs>
    </Input>
  </DataInputs>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>GeoTiff</ows:Identifier>
      <ows:Title>Interpolated Output Map</ows:Title>
      <ows:Abstract>Resulting interpolated map. With X,Y coordinates, predicted
value (var1.pred), standard-deviation (var1.stdevev) and variance (var1.var), for each
interpolated point, as a link to a stored geotiff. The different results will be allocated to
different bands: band1 == var1.pred, band2==var1.stdev and
band3==var1.var</ows:Abstract>
      <ComplexOutput defaultFormat="image/tif">
        <SupportedComplexData>
          <Format>image/tif</Format>
        </SupportedComplexData>
      </ComplexOutput>
    </Output>
  </ProcessOutputs>
</ProcessDescription>
</ProcessDescriptions>
```

The choice of the input/output type is based on the use of GDAL (Geospatial Data Abstraction Library) as support library in the python script and on the simplicity with which a table with x, y, z values in a comma separated value format (CSV) can be converted to GML using the *ogr2ogr* command line. The GML structure is transferred with the data to and from RAISIN using a standard WPS procedure, where the GML is embedded as *ComplexValue* in the XML document (Table 5 contains an example of execute request with two GML points as data input).

Table 6: Execute result of *AutoKrigingProcess* with a complex value reference indicating the location of the interpolation result as a GeoTIFF.

```
<?xml version="1.0" ?>
<ExecuteResponse version="0.4.0" xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wps
http://www.ogcnetwork.net/schemas/wps/0.4.0/wpsExecute.xsd">
<ows:Identifier>autokrigeprocess</ows:Identifier>
<Status><ProcessSucceeded/></Status>
<ProcessOutputs>
<Output>
<ows:Identifier>GeoTiff</ows:Identifier>
<ows:Title>Interpolated Output Map</ows:Title>
<ows:Abstract>Resulting interpolated Map. With X, Y coordinates, predicted
value (var1.pred), standard-deviation (var1.stdev) and variance (var1.var), for each
interpolated point, as a link to a stored GeoTIFF. The different results will be allocated
to different bands: band1 == var1.pred, band2==var1.stdev and
band3==var1.var</ows:Abstract>
<ComplexValueReference encoding="utf-8" format="image/tif"
ows:reference="http://remwps.jrc.it/wpstmp/interpolation176304.tif" schema=""/>
</Output>
</ProcessOutputs>
</ExecuteResponse>
```

The *AutoKrigingProcess* response is a XML document containing a reference for the location of the GeoTIFF containing the interpolation result in an image format. This reference is indicated in the *ComplexValueReference* tag. Geotiff contains the interpolation values in the form of raster pixel points in band 1 and the standard-deviation and variance in bands 2 and 3, respectively. The GeoTIFF file is stored in the process server and can be



downloaded by means of a HTTP-GET request. This can be, for example, simply done by writing the URL in a browser (Table 6).

## 5 DISCUSSION

The purpose of this paper was to present a web service for the real-time interpolation of environmental data and to discuss its potential applications. The development of REM's Automatic Interpolation Service for INTAMAP (RAISIN) showed how to port interpolation functions to a SOA model structure, using WPS as the interchange method and LAPR as the bundle software with R's `automap` package.

RAISIN is designed to be flexible and portable since all developments are "open source", allowing the service to run on any computer platform and this with little hardware requirements. The use of the SOA/WPS model further allows the system to be integrated in any decision support system or monitoring platform. Other model possibility like SOA/WFS (Web Feature Service) could be implemented, but it would need more operations to achieve the same final result of RAISIN and the input/output would be more GML dependant leaving few or little flexibility.

The limitations of the RAISIN fall in two categories: statistical and technical ones.

In a statistical point of view, the generic use of the system for the real-time mapping of environmental data is currently limited to cases in which the residuals of the input data can be described by a stationary random field, something that is frequently not verified, especially in the case of local anomalies that can destruct the spatial structure of the experimental semivariogram (see e.g. Pebesma 2005).

On the technical level, the use of XML can be a bottleneck in the efficiency of the system since it is very verbose and consumes a considerable amount of resources. It is indeed inefficient to send an XML file with thousands of point observations, even if the response does not have such limitation because it contains only the location of a raster image in a binary format (which is more efficient for storing data when compared to a string structure). The use of GML as the input standard defines how spatial data are organized and provides a standard for embedding the data in the XML request. However, the GML standard format allows the same data to be described in different ways, creating problems when parsing the data to the `wps.py` script. Requesting from the user to use the GDAL libraries may not always be possible and RAISIN should therefore also provide independent

processes of data transformation for facilitating the use of the *Auto-KrigeProcess*.

Future developments will focus in the short term on porting RAISIN to the standard WPS 1.0.0. Long-term efforts may lead to the use of universal kriging rather than ordinary kriging as implemented here. Information on interpolation errors (using e.g. cross-validation statistics) will also be automatically provided to end-users. Not less ambitious is the work to be done on dealing with the uncertainties that are associated with the information exchanged. The possible use by RAISIN of UncertML, an XML schema for describing uncertainty that is currently in development at the Aston University (Williams et al. 2007), would be a major step in this direction. The purpose of UncertML is to allow end-users to better manage information on the uncertainty that is associated to the information exchanged with the web service, and consequently to better understand error propagation in the modelling process. This issue is fundamental in environmental risk assessment.

## 6 ACKNOWLEDGEMENT

This work is funded by the European Commission, under the Sixth Framework Programme, by the Contract N. 033811 with the DG INFSO, action Line IST-2005-2.5.12 ICT for Environmental Risk Management. The views expressed herein are those of the authors and are not necessarily those of the European Commission.

More information on RAISIN can be found in the web pages of INTAMAP ([www.intamap.org](http://www.intamap.org)).

## 7 REFERENCES

- Brenning, A. and G. Dubois (2008). "Towards generic real-time mapping algorithms for environmental monitoring and emergency detection". Stochastic Environmental Research and Risk Assessment (SERRA). Vol. 22(5). In Press.
- Chiles, J., and P. Delfiner (1999). Geostatistics: Modeling Spatial Uncertainty. John Wiley & Sons, New York.
- Cressie, N. A. (1993). Statistics for spatial data. Wiley, New York.
- EUR (2003). Mapping radioactivity in the environment. Spatial Interpolation Comparison 1997. Dubois G., Malczewski J. and De Cort M. (Eds.) EUR 20667 EN, Office for Official Publications of the European Commission, Luxembourg, 268 p.

- EUR (2005). Automatic mapping algorithms for routine and emergency monitoring data. Report on the Spatial Interpolation Comparison (SIC2004) exercise. Dubois G. (Ed.). EUR 21595 EN. Office for Official Publications of the European Commission, Luxembourg, 150 p.
- Goovaerts, P. (1997). Geostatistics for natural resources evaluation. Oxford University Press, New York.
- Hiemstra, P., E. Pebesma, C. Twenhofel and G. Heuvelink (2008). "Real-time automatic interpolation of ambient gamma dose rates from the Dutch Radioactivity Monitoring Network." Computers and Geosciences, submitted
- Izza, S. L. Vincent and P. Burlat (2008) "Exploiting semantic web services in achieving flexible application integration in the microelectronics field." Computers in Industry. In press
- Jarvis, C. H., N. Stuart and W. Cooper (2003). "Infometric and statistical diagnostics to provide artificially-intelligent support for spatial analysis: the example of interpolation." International Journal of Geographical Information Science 17(6): 495-516.
- Open Geospatial Consortium Inc. (2002). OpenGIS Geography Markup Language (GML) Implementation Specification, version 2.1.2 (OGC 02-069). Editors: Simon Cox, Adrian Cuthbert, Ron Lake & Richard Martell.
- Open Geospatial Consortium Inc. (2005). OpenGIS Web Processing Service (OGC 05-007r4g). Editors Peter Schut & Arliss Whiteside.
- Pebesma, E. J. (2004). "Multivariable geostatistics in S: the gstat package." Computers and Geosciences 30(7): 683–691.
- Pebesma, E. J. (2005). "Mapping radioactivity from monitoring data: Automating the classical geostatistical approach." Applied GIS 1(2): DOI:10.2104/ag050011
- Pebesma, E.J., Dubois, G. and D. Cornford (2008). "The challenge of real-time automatic mapping for environmental monitoring network management". In: "geoENV VI – Geostatistics for Environmental Applications" Quantitative Geology and Geostatistics. Soares, A., Pereira, M. J., Dimitrakopoulos, R. (Eds.), pp. 467-476. Springer Verlag.
- R Development Core Team (2008). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org>

Williams, M., D. Cornford, L. Bastin and B. Ingram (2007). UncertML:  
Overview and High Level Architecture. V.1.1.  
<http://www.intamap.org/pub/UncertML.pdf>