



Proposal/Contract no.: 033811
Project start: September 1, 2006
Project end: August 31, 2009



INTAMAP

Interoperability and Automated Mapping

<http://www.intamap.org/>

SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2005-2.5.12
ICT for Environmental Risk Management

Deliverable 6.2.

*Draft validation plan for the automatic mapping web service,
including specification of test data*

Reference	
Title of Deliverable	Draft validation plan for the automatic mapping web service, including specification of test data
Related Workpackage and Tasks	WP6, Task 6.2.
Type of Document	Public
Authors	JRC, BfS, AST
Date	28 November 2007
Version	2.1

Project coordinator

Prof.dr. Edzer J. Pebesma
Utrecht University, The Netherlands
E-mail: e.pebesma@geo.uu.nl

REVISION RECORD

Version	Date	Changes	Authors
Version 1	28 August 2007	First version	G. Dubois, J. de Jesus, U. Stöhlker
1.1	28/8/2007	Edited, and added detail to system requirements	Dan Cornford
1.2	28/8/2007	Edited	G. Dubois
1.3	02/10/2007	Edited, addition of figure, file formatting	G. Dubois, J. de Jesus
Version 2	14/11/2007	Major restructuring to fulfil the req. for EU acceptance. Change in chapter structure.	J. de Jesus, B. Doherty
2.1	28/11/2007	Edited, inclusion of executive summary, addition and improvement of test plans	J. de Jesus
2.2	29/11/2007	Fixed some definitions, added acceptance tests.	Dan Cornford

Executive Summary

Title in proposal: Validation plan for automatic mapping.

Description in proposal: Draft validation plan for the automatic mapping web service, including specification of test data.

Summary of deliverable

This deliverable specifies the validation plan we will adopt across the INTAMAP project, especially in WP1 and WP6. We describe the components of the validation plan which include an automatic validation sever (rather like a web service based regression / unit testing service), formal tests of correctness for the implemented web service, and user acceptance testing. The characteristics of the independent web service that will validate and quantify the quality of the interpolation, store validation data sets (scenarios) specified by BfS (see Deliverable D5.3) that will be used for validation is defined.

The report covers and defined the guidelines for assembling and running a validation web service. The functional system requirements (second section), defines basic guidelines for the validation web service based on ISO 9126 and how the validation web service will interact with the users and the interpolation server being tested. In the third section we define the technical details of the server, the languages (Python and R) and software (Apache) used and the major web service standard (WPS). The decision on technical details reflects the open source philosophy of INTAMAP ensuring compatibility among different deliverables.

The fourth section provides 22 formal validation plans to be implemented by the validation server, the plans are contained in 5 independent groups: Technical support, Data set requests, Statistical Tests, Advanced statistical and Stress tests. These set of tests should ensure correct functioning of the system, measure the response of the system to stress and quantify the accuracy of the interpolation service. This validation service will be employed in conjunction with user acceptance testing for which we will engage with the advisory panel currently in formation in response to the first year review.

Legal Notices

The information in this document is subject to change without notice.

The Members of the INTAMAP Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the INTAMAP Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

TABLE OF CONTENTS

Executive Summary	3
Summary of deliverable	3
I. Introduction	6
1.1 Objectives of the Validation Plan	6
II. Functional System Requirements	7
2.1 Introduction	7
2.2 General Testing Requirements.....	7
2.2 Validation User Client.....	8
2.3 Test Data Sets.....	9
2.4 Reports	10
2.5 Validation Server	10
2.6 Interpolation server	10
2.7 User acceptance testing	11
III. Technical Requirements and Considerations	12
3.1 Introduction	12
3.2 Technical Requirements for Web Service	12
3.3 Input and Output Processing.....	13
3.4 Web Server	13
3.5 Python	13
3.6 R statistical environment.....	14
IV. Validation Plans.....	15
4.1 Plan A – Technical Correctness	16
4.2 Plan B – Data set requests / User data set processing	17
4.3 Plan C - Statistical Test	19
4.4 Plan D – Advanced statistical tests.....	21
4.5 Plan E - Stress Tests.....	23
V. Final Considerations and Summary	24
References.....	25

I. Introduction

The maps produced by INTAMAP's interpolation web service will have an independent quality control service that will verify the quality of the generated grids, according to a validation plan that quantifies the interpolation error using a number of scenarios and/or test data. The validation plans are pre-defined tests made by an independent web service to the interpolation web service, following the client-server philosophy and allowing for platform independence.

1.1 Objectives of the Validation Plan

The major objectives of the validation plan / validation tests are:

- Show how we intend to ensure the correctness of the software implementation;
- Determine the algorithmic accuracy of the interpolation service (the accuracy will be reported as a numerical values);
- Determine the efficiency of the interpolation service for several scenarios;
- Store data-sets from different scenarios;
- Quantify the time efficiency of the interpolation service (benchmarking);
- Determine statistical error parameters of user's data-sets (mean error, standard deviation, etc.) submitted to the server;
- Describe the user acceptance testing envisioned.

II. Functional System Requirements

This section reviews the major requirements for validating systems we will develop in INTAMAP, but does not focus heavily on the capture of user requirements, which are described in Deliverable D1.1.

2.1 Introduction

The validation service is a secondary system that will support the interpolation web service of INTAMAP. The functions provided and the operations to be executed are based on algorithms and functions commonly used in geostatistical studies. As base for the validation plan it will be used the experience gathered in the interpolation exercise of SIC97 and SIC2004 where the participants were invited to use several interpolation strategies/methods and later their interpolated results were validated with the real values. The description of the validation system will follow figure 1.

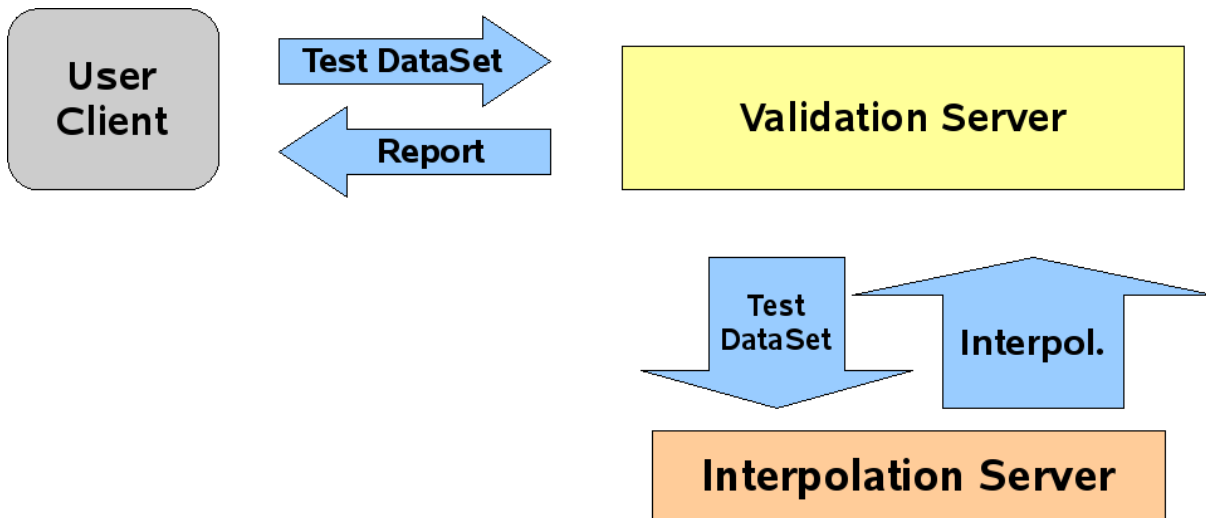


Figure 1. General structure of the validation web service. The system is composed of three “players”, the user client that requests the validation, the validation server that will provide the validation web services and the interpolation server to be tested.

2.2 General Testing Requirements

The validation template for the interpolation web client will follow some of the guidelines of ISO 9126 (standard for software testing and validation).

The testing procedure should indeed consider functionality, reliability and usability/efficiency

Functionality

It should check whether the services and functions provided by the interpolation web service are compliant with the necessities of spatial interpolations.

Reliability:

The reliability of the web interpolation service should be tested under several conditions:

- Time of reply without a specific web service request (ping);
- Time of reply for processes execution (for example data submission);
- Time of reply when processing a previously web request;
- Check for server reply when a fault is generated in the interpolation web service.

Usability/Efficiency:

The web interpolation service should provide interpolated grids in the most efficient and accurate way. During the Spatial Interpolation Comparison Exercise (SIC) 2004 (EUR 2005), a list of statistical criteria was defined to compare the efficiency of the interpolation algorithms (Myers, 2005):

- Correlation between v_i and v_i^*
- Correlation between $(v_i - v_i^*)$ and v_i
- Mean Error, $ME = (1/n) \sum_{i=1, \dots, n} [v_i - v_i^*]$
- Mean Absolute Error, $MAE = (1/n) \sum_{i=1, \dots, n} |v_i - v_i^*|$
- Mean Square Error, $MSE = (1/n) \sum_{i=1, \dots, n} [v_i - v_i^*]^2$
- Mean Normalized Square Error, $MNSE = (1/n) \sum_{i=1, \dots, n} \{ [v_i - v_i^*] / \sigma_i \}^2$

Where n is the number of observations, v_i the observed value at location i and v_i^* the estimated value at the same location, and σ_i^2 is the Kriging variance at location i .

In addition to these criteria, misclassification rates need also to be assessed, as too many false negatives would quickly discredit the potential use of an automatic system for emergency purposes. Therefore, Mahalanobis distances and ROC (Receiver Operating Characteristics) curves will also be used for quantifying misclassification rates, this constitutes a specific subset of tests connected with classification of values.

ROC in particular is used to plot the probabilities of detection against the probabilities of having false alarms (see e.g. Swets, 1988; Goovaerts, Jacquez, and Marcus, 2005). Useful to summarize the graphical form of the ROC is the Area Under the Curve (AUC) statistics that is used as a scalar measure: an AUC of 0.5 reflects a random classification, while an AUC of 1 implies a perfect match.

2.2 Validation User Client

The validation web service will follow the WPS (Web Processing Service) specifications defined by the Open GIS Consortium (OGC) and INSPIRE. Consequently, the validation web service will have the ability to indicate the services it provides, how to run them, and other necessary metadata.

Following the philosophy above described, a client to the validation service can be a person that requests any of the services provided by the service, a web client that requests a service on the behave of a person and another web service that needs a validation service in its own provided services (chained processes) .

2.3 Test Data Sets

The validation server will contain specific scenarios and associated datasets as defined by BfS (see deliverable 5.3 for additional information):

- Scenario 1: Front carrying rain over Europe
- Scenario 2: Helicopter Based Measurements
- Scenario 3: Medium Scale Simulation
- Scenario 4: Large Scale Simulation 1 (Offenbach release)
- Scenario 5: Large Scale Simulation 2 (12°7' E, 57°16' N release)

Each dataset will contain N known measurements.

The client will be able to request the totality of the data sets or a reduced sub-set (n), but the major objective is not for simple data set storage/retrieval but use in a validation procedure.

For testing a validation service the user will have to craft a proper request to the validation service with the following parameters:

- Desired scenario
- Number of points to be interpolated (n), where $n < N$
- If n will be randomly picked from N or will constitute a spatial subset of N
- Web location of the interpolation server

After receiving the proper execution request, the validation service will create the request to the interpolation web service which will upload a subset of the chosen dataset. The uploaded data will thus contain the $(N-n)$ data points and the interpolation at the n remaining points will be requested.

The reply of the interpolation server will be returned to validation server and then processed and returned to the client (the validation server will be working as a proxy).

To extend the functionality of the validation service, (with the object of porting it to other field) the user will also have the possibility of passing specific data sets to the validation service. In this case the user will have to upload the major data set (N) and it's (n) points to be interpolated, as explained for the stored scenarios

2.4 Reports

The reports will follow the OGC's philosophy of XML output response, following the WPS specifications. The output report will change according to the user client request, for example it may contain numerical values of statistical criteria that define interpolation efficiency and accuracy (SimpleValue), data sets from stored data (ComplexValues). The XML output report response will allow for human and server clients to process the results with relative ease.

2.5 Validation Server

The validation server comprehends the following structures:

- Server for accepting, replying and chaining requests
- Database for data set storage and manage

The validation server is independent of any interpolation service/server, this independence is important since INTAMAP has also the objective of providing generic services than can be ported to other situations. In principle this independence will allow for the validation server to participate in any chaining service that will need some of the processes provided by the validation server.

The server will provide processes that will work as specific plans (see chapter IV for detail plan description)

The test data sets (scenarios) will be stored in a database, for a more efficient service. The test data sets will only be available thru a specific WPS request, and not by direct database access.

The validation server will not be limited by the data sets (scenarios) it can provide for validation purposes, and indicated in Figure 1, data sets can "flow" from the user client to the validation server and finally to the interpolation service. Therefore the validation server will accept data sets in pre-defined formats that will be sent to the interpolation server and used for validation purposes. The adoption of user defined data-sets allows for the validation server to be independent of the phenomena being studied/interpolated.

The validation service can be considered a "chained service" since it will have to make WPS requests to the interpolation servers, therefore the majority of the validation plans will require the URL or IP of the interpolation service.

2.6 Interpolation server

The interpolation server is a totally independent structure with its own processes offerings, input/outputs and reports, running independently and not depending in any processes offered by the validation server. The interpolation server of INTAMAP is the major task of WP 1

The validation server should be flexible enough to make requests, interpreter responses, agree on version and data formats to be used.

2.7 User acceptance testing

The first year review meeting for the INTAMAP project stressed the importance of users in the validation process. To this end the INTAMAP team are in the process of developing an advisory panel of potential users (these have will also have input into D1.1 as the requirements evolve in response to changes in technology and science). The advisory panel will consist of application experts who might utilise a system such an INTAMAP intends to develop, and will thus make excellent testers for user acceptance of the system. To this end the advisory panel will be asked to complete a user acceptance testing process, including an interview and questionnaire. This process will be undertaken across a range of time scales as prototypes are developed to a sufficient level for exposure to the advisory panel, in line with the unified process development methodology we employ in the project. Thus the acceptance testing will feed into further developments of the systems; the web client, the validation service and the interpolation service.

III. Technical Requirements and Considerations

3.1 Introduction

The system that will perform the validation plan has many aspects in common with the interpolation service. While some technical options need to be considered as absolute requirements, others can be taken only as recommendations. To ease portability and developments, communication protocols and data formats should follow mandatory requirements. On the other hand, “internal” software processing can follow looser recommendations.

3.2 Technical Requirements for Web Service

The major web services will be provided using Open Geospatial Consortium's Web Processing Service (WPS).

WPS provides clients with the access across a network to pre-programmed calculations, computation models and functions that operate on spatially referenced data. Calculations can, in theory, be of any type and of any complexity and take any number of inputs and outputs. The data delivered as input/output can be in a GML (Geographic Markup Language) standard or image data format like GeoTIFF.

The WPS follows a logic based on three operations: `GetCapabilities`, `DescribeProcess` and `Execute`. Using only these operations and other options, information can be retrieved on the server with the supplied processes and the required process executed.

The encoding of the requested operations follows a KVP (Key Value Pair) structure. For example, a request of `GetCapabilities` using HTTP and GET could be as follows:

<http://validation.jrc.it/service=WPS&request=GetCapabilities&AcceptVersion='0.0.4'> .

The use of XML (Extensible Markup Language) is also possible when making WPS requests.

The `GetCapabilities` operation of WPS allows clients to retrieve service metadata from the server. The response will be in the form of a XML document containing service metadata about the server and a brief process description. The document can be used by a web client to “understand” the general characteristics of the validation service and used by a GUI (Graphical User Interface) system to dynamically display the validation options that are available to the end user.

`DescribeProcess` allows for clients to request and receive back a full description of one or more processes that can be executed by the `Execute` operation. It describes the input/output formats and tells if the server supports storage and/or status of operation. The `DescribeProcess` operation can be done using KVP or an XML specially crafted for the purpose.

`Execute` is the most important operation of the WPS as the input data are here uploaded to the server, processed and the output returned. Several options of the `Execute` operation are essential for the success of the validation process: the `Storage` option, for example, can be a useful means to store specific results that can be used later on. One option that should be activated when the execution request is made is the `Status` one. This option will activate a reply on the initial status of the operation and an URL (Universal Resource Location) link will provide information on the level

of the completion of the process.

3.3 Input and Output Processing

The datasets transferred to the interpolation web service will be in GML (v. 3.1.1 or v 2.2.1) format, which in case of need will be supplemented with SWE (Sensor Web Enablement) and Observation & Measurement schemes. The transferred file will have a simple point data structure (X,Y,Z), with X and Y being the coordinates in 2D and Z the measurement value associated to the location.

The reply of the interpolation web service will be a XML document containing the interpolation results embedded in the document or a web resource locator for the interpolated result. Additional information will be contained as LiteralValues (WPS specification of data reply) strings with numerical values. The validation server will then extract the information from this initial XML document to evaluate the interpolation web service. After evaluating the result, the web client will produce a XML document containing statistical results and other information on the tested parameters as LiteralValues strings with numerical values. This will be the reply provided by WPS's `Execute` request made from the client to the web validation service

3.4 Web Server

The web server structure for providing the validation services will be built around the apache server software (<http://httpd.apache.org/>). The apache server will run on a server using Linux as the operating system.

The web server will be responsible to provide HTTP (port 8080) standard connections and also HTTPS (port 443) encryption connections. This should be sufficient to deal with GET/POST HTTP requests necessary for the provided web services (mainly validation and other mathematical services as web services).

The security of the web server is of major concern and some basic guidelines will have to be followed.

- No CGI execution will be allowed;
- No insecure modules like Server Side Include and minimal use of features/modules will be accepted. No status module or server type/information will be passed
- No .htaccess overwrite allowed
- Use of security module (mod_security) for traffic control
- Jailing of the server and the R libraries is needed

3.5 Python

Python is a high level programming language released under GPL License (Gnu General Public License, normally referred as Open Source). This language provides a broad standard library, providing tools to address several sorts of tasks. Python is integrated into the Apache server as a module, providing a handle to deal with user requests to the server. When a request is made for a specific task a Python script is called to deal with it.

A XML (Extensible Markup Language) or GET/POST service request is processed by a Python script that will provide with the proper reply according to the service being implemented. The script may implement other functions based on R or even run code in other languages like C++ or MATLAB.

Currently there is a Python implementation for WPS, known as PyWPS and supported by the DUB (Deutsche Bundesstiftung Umwelt), this implementation will be used as support for the validation server WPS.

3.6 R statistical environment

R is a free software environment for statistical computing and graphics which can compile under a variety of platforms (see <http://www.r-project.org/>). Many functions have already been structured in packages that can be downloaded and integrated within the R environment.

R can run through an interactive prompt, as a script or even inside other languages by means of specific interfaces (e.g. TCP/IP or SOAP [Simple Object Access Protocol]) or of R-libraries (i.e. C/C++, Python, Perl).

In the case of the validation web service, the `RPY` module of Python will be used to access R's functions and data objects for data validation/processing. This approach implies that part of the server's development will be done by development specific functions/scripts do deal with data validation/processing.

IV. Validation Plans

The validation plans provided by the validation server are WPS processes. There are 5 major types of plans that will be provided:

Plan A – Technical support

Plan B – Data set request (BfS scenarios) / User data set processing

Plan C – Statistical tests (ratios and correlations)

Plan D – Advanced statistical tests (Mahalanobis distances, Type I and II errors, ROC)

Plan E – Stress tests (Benchmark of interpolation servers)

The 5 group of plans have several subsections that are going to be described using standard OGC's document format for tests.

4.1 Plan A – Technical Correctness

Plan A.1 - Interpolation server availability

- a) Objective: Determine if the server and WPS is online.
- b) Test Input: URL or IP location of the interpolation server.
- c) Test Method: Ping (ICMP) of Interpolation server.
- d) Test Output: WPS XML response with 4 ping times.

Plan A.2 – Port Scanning

- a) Objective: Determine if ports 80 (HTTP) and 443 (HTTPS) of validation server are open.
- b) Test Input: URL or IP of the interpolation server.
- c) Test Method: Port scan using TCP three way handshake.
- d) Test Output: WPS XML response with port status.

Plan A.3 – DNS to IP conversion and vice-versa

- a) Objective: Determine the DNS or IP of interpolation server. It is preferred for the validation server to use direct IP requests.
- b) Test Input: Hostname or IP of the interpolation server.
- c) Test Method: DNS lookup (Hostname to IP) or reverse DNS (IP to Hostname).
- d) Test Output: WPS XML response with IP or Hostname.

Plan A.4 – Capacities of interpolation server

- a) Objective: Determine if server provides with interpolation service.
- b) Test Input: URL or IP location of the interpolation server.
- c) Test Method: WPS 's `GetCapabilities` and `DescribeProcess`.
- d) Test Output: Text report in WPS XML response grouping the type of interpolations provided (deterministic and stochastic), Input/Output of interpolations, time of interpolation server for reporting results.

4.2 Plan B – Data set requests / User data set processing

Plan B.1 – Complete download of data set

- a) Objective: Download specific data sets stored in the server (BfS scenarios).
- b) Method: Simple scenario request.
- c) Input: Number of scenario, reply type: GML in XML response or stand alone GML.
- d) Output: Complete data set of scenario either in the WPS XML response (GML embbebed as ComplexValue) or download file in GML format.

Plan B.2 – Download of partial data-set

- a) Objective: Download part of data sets stored in server (BfS scenarios).
- b) Method: Creation of a sub-set (A) from the scenario data set (B).
Therefore: $(A \in \mathbb{R}^n) \subset (B \in \mathbb{R}^n)$.
- c) Input: Number of scenario, reply type: GML in XML response or stand alone GML, number of points to be used.
- d) Output: subset A either in the WPS XML response (GML embbebed as ComplexValue) or as a download file in GML format.

Plan B.3 – Download of validation of sub dataset

- a) Objective: Download of specific data sets to be used in interpolation and later cross-validation.
- b) Method: Download of a data set with total data features (A) and data set (B) with only coordinates (unknown Z values in a X,Y,Z structure), both sets have the same number of points. A 3rd data set (C) will contain the remaining points from the scenario (D).

Therefore: $(B \in \mathbb{R}^2) \subset (A \in \mathbb{R}^3)$, $|A| = |B|$

$$A \cup C = D \wedge A \cap C = \emptyset$$

- c) Input: Number of scenario, reply type: GML in XML response or stand alone GML, number of points in set A/B.
- d) Output: sub sets A, B and C in the WPS XML response (GML embbebed as ComplexValue) or download file in GML format.

Plan B.4 – Validation sub sets from user

- a) Objective: To facilitate the creation of validation sub sets for validation proposes of plans C.1b, C.2b,C.3b,C.4b.
- b) Method: Upload of data to validation server, identical steps as plan B.3 but using the uploaded data as scenario (set C).
- c) Input: GML file with data, number of points in set A/B.
- d) Output: sub sets A, B and C in the WPS XML response (GML embbebed as ComplexValue) or download file in GML format.

4.3 Plan C - Statistical Test

Plan C.1a – Simple Correlation

- a) Objective: Simple correlation between v_i and v_i^* (v_i the observed value at location i and v_i^* the estimated value at the same location).
- b) Method: The validation server will make an internal WPS request using plan B.3, getting sub set A,B and C (A is a sub set of C and B only contains X,Y points). The validation server will send set C as data and B as interpolation request. The returned B set (with interpolated values) will be compared (simple correlation) with set A that was previously stored.
- c) Input: URL or IP of the interpolation server, scenario to be used, number of points for set A/B.
- d) Output: WPS XML response with a and b parameter of correlation equation and also r^2 .

Plan C.2a – Relative Correlation

- a) Objective: Correlation between $(v_i - v_i^*)$ and v_i (v_i the observed value at location i and v_i^* the estimated value at the same location).
- b) Method: Same as Plan C.1a but with final calculation of (A-B).
- c) Input: URL or IP of the interpolation server, scenario to be used, number of points for sub set A/B.
- d) Output: WPS XML response with a and b parameter of correlation equation and also r^2 .

Plan C.3a – Combined Correlation

- a) Objective: Simultaneous correlation (Combination of Plan C.1 and C.2).
- b) Method: Same as Plan C.1a and C.2a.
- c) Input: URL or IP of the interpolation server, scenario to be used, number of points for set A/B.
- d) Output: XML response with a and b parameter of correlation equation and also r^2 , for each type of correlation.

Plan C.4a - Mean Error Tests

- a) Objective: Determine the mean error, mean absolute error, mean square error and mean normalized square error.

b) Method: Same as Plan C.1a but with final calculations according to the parameter list:

- Mean Error, $ME = (1/n) \sum_{i=1, \dots, n} [v_i - v_i^*]$
- Mean Absolute Error, $MAE = (1/n) \sum_{i=1, \dots, n} |v_i - v_i^*|$
- Mean Square Error, $MSE = (1/n) \sum_{i=1, \dots, n} [v_i - v_i^*]^2$
- Mean Normalized Square Error, $MNSE = (1/n) \sum_{i=1, \dots, n} \{ [v_i - v_i^*] / \sigma_i \}^2$

where n is the number of observations, v_i the observed value at location i and v_i^* the estimated value at the same location, and σ_i^2 is the Kriging variance at location i .

c) Input: URL or IP of the interpolation server, scenario to be used, number of random points OR coordinate box of sampling area.

d) Output: WPS XML response with the values of the parameters described in method.

Plans C.1b, C.2b, C.3b, C.4b

a) Objectives: Identical to their respective contra-parts (C.1a, C.2a, C.3a, C4a).

b) Method: The user will provide the data sets for validation. This could be done automatically using plan B.4.

c) Input: URL or IP of the interpolation server, proper data set (output of plan B.4).

d) Output: Identical to their respective contra-parts (C.1a, C.2a, C.3a, C4a).

4.4 Plan D – Advanced statistical tests

Plan D.1 – Mahalanobis distance

- a) Objective: Calculate the Mahalanobis distance between a set of withheld points (A) and the interpolation service prediction given by the mean and covariance (μ , P), derived from the observed part of the data set (B).
- b) Method: The user will provide data set A and B. The validation server will obtain the mean, μ , and covariance matrix P from the interpolation service, passing it B as the observations and the locations of A as the prediction points and use the following formula to calculate the Mahalanobis distance $D_M(x)$:

$$D_M(x) = \sqrt{(x - \mu)^T P^{-1} (x - \mu)}$$

- c) Input: Data set A and B.
- d) Output: WPS XML response with μ , P and $D_M(x)$.

Plan D.2 – Confusion Matrix

- a) Objective: Calculation of False Positives (FP) (Error Type I) and False Negatives (FN) (Error Type II), and creation of confusion array (or matrix) with: FP, FN and True Positives (TP), True Negatives (TN).
- b) Method: The calculations are based on the delimitation of an “abnormal area” ($A \in \mathbb{R}^3$) from the total spatial domain ($B \in \mathbb{R}^3$), therefore ($A \subset B$) and defining a specific threshold value (x). The calculations are then performed for each member point (p) of A and B.

False Positive == Point contained in A that is below the threshold ($p \in A \wedge p < x$)

True Positive == Point contained in A that is above the threshold ($p \in A \wedge p > x$)

False Negative == Point contained in B that is above the threshold ($p \in B \wedge p > x$)

True Negative == Point contained in B that is below the threshold ($p \in B \wedge p < x$)

- c) The final stage will be the counting of the number of FP, FN, TP and TN points (confusion array).
- d) Input: GML file with points and polygon of “abnormal” area(s) as feature, threshold value.
- e) Output: WPS XML response with confusion array.

Plan D.3 - Confusion Array ratios

- a) Objective: Calculation of ratios associated with confusion array (Plan D.2).
- b) Method: Use of Plan D.2 and posterior calculation of :

- i. True Positive Rate (TPR) = $TP / (TP+TN)$
- ii. False Negative Rate (FPR) = $FP / (FP+TN)$
- iii. Specify (SPC) = $1 - FPR$
- iv. Accuracy (ACC)= $(TP+TN)/(TP+TN+FP+FN)$

c) Input: Same as plan D.2.

d) Output: Same as plan D.2 plus calculated ratios.

Plan D.4 – ROC (Receiver Operating Characteristics) and AUC (Area Under the Curve) calculation

- a) Objective: Calculation of the ROC curve and determination of the area below the curve (AUC).
- b) Method: Calculation of probability of TP and FP, using a Gaussian distribution centred on the threshold value. Plan D.3 to make TP and FP calculations. Points for ROC will be calculated and the area under the curve of points will be calculated.
- c) Input: Same as Plan D.2.
- d) Output: WPS XML response with array of points for the ROC curve and the value of AUC.

4.5 Plan E - Stress Tests

Plan E.1 – Time to interpolate a specific number of points

- a) Objective: Benchmarking the efficiency of interpolation.
- b) Method: Determining network latency using plan A1. Requesting a N point interpolation for one of the scenarios, using plan B.3. Monitoring time necessary between request and reply.
- c) Input: Number of scenario, number of points.
- d) Output: WPS XML response with number of points interpolated, unix time for start and end of process, total time for processing request, network latency.

Plan E.2 – Increasing stress

- a) Objective: Graphic benchmarking of interpolation efficient.
- b) Method: Identical to Plan E.1. Loop request, with $N+\Delta$ points (Δ – increase defined in input).
- c) Input: Number of scenario, initial number of points, final number of points, number of cycles in loop.
- d) Output: WPS XML response. For each number of points interpolated: unix time for start and end of process, total time for processing request, network latency.

V. Final Considerations and Summary

This document describes the validation processes that will be applied in the INTAMAP system. It should be read with D1.1 (user requirements) in mind. The document sets out the range of validation tests that will be applied to the INTAMAP service, ranging from formal tests of correctness of the implementation, through statistical tests on the results of the interpolation requests to user acceptance testing of the final system. In order to simplify the testing, and in line with the unified process development methodology employed within the project, a validation service (often referred to as a client) will be built to automate much of the testing. This can be seen as a service oriented architecture version of unit and regression testing.

The web validation client will use pre-determined statistical criteria for interpolation checking and accuracy under specified scenarios. As a draft validation plan it was defined 22 tests that will be implemented to assure the quality level of the interpolation server, these validation plans will be increased and improved in the future to better suit the objectives of INTAMAP. The INTAMAP clients and web service will also undergo user acceptance testing to ensure they are relevant to the communities we hope will take up the INTAMAP solution after the end of the project.

References

EUR (2005). *Automatic mapping algorithms for routine and emergency monitoring data. Report on the Spatial Interpolation Comparison (SIC2004) exercise*. EUR 21595 EN, Dubois G. (Ed), Luxembourg, 150 p.

Swets JA (1988) Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293.

Goovaerts, P., Jacquez, G.M., and W.A. Marcus (2005). Geostatistical and local cluster analysis of high resolution hyperspectral imagery for detection of anomalies. *Remote Sensing of the Environment*, 95: 351-367.

Myers D.E. (2005) Spatial Interpolation Comparison exercise 2004: a real problem or an academic exercise? In: *Automatic mapping algorithms for routine and emergency monitoring data*. EUR 21595 EN, Dubois G. (Ed), European Commission, Luxembourg, pp 79-88