



DELIVERABLE D1.4.1

WP1, Task1.4.1, Weather forecasting and air pollution modelling

Document Filename:	CG-1.4-DOC-ICM004-SRS
Work package:	WP1 Interactive and data intensive applications
Partner(s):	ICM, USC, CSIC
Lead Partner:	ICM
Config ID:	CG-1.4- DOC-ICM004-2-0-DRAFT-A
Document classification:	Public

Abstract: The document provides a description of the initial specification and the requirements for Grid tools needed in task 1.4 of the WP1 "Weather forecast and air pollution modelling". This meteorological application in the CrossGrid project includes three different subtasks. The first one is concentrated on sea wave modelling forced by the winds simulated from a Coupled Ocean/Atmosphere Mesoscale Prediction System. The second one covers data mining techniques on distributed databases in the Grid environment applied to downscaling weather forecasts, and the third one is focused on the implementation of air-pollution models based on the STEM-II application.



Delivery Slip

	Name	Partner	Date	Signature
From	WP1, task 1.4	ICM, USC, CSIC	03.05.2002	
Verified by				
Approved by				

Document Log

Version	Date	Summary of changes	Author(s)
1-0-DRAFT-A	12 Mar 02	Draft version	Bogumil Jakubiak
2-0-DRAFT-A	14 mar 02	Improved draft version	Bogumil Jakubiak, Francisco F. Rivera
3-0-DRAFT	25 Mar 02	Next improvements	Francisco F. Rivera
1-0-DRAFT-A	03 May 02	Joined draft version	Bogumil Jakubiak, Francisco F. Rivera, Antonio Cofiño, Jose Manuel Gutierrez, Jesus Marco, David Rodriguez
2-0-DRAFT-A	13 May 02	Improved version	Bogumil Jakubiak, Francisco F. Rivera, Antonio Cofiño, Jose Manuel Gutierrez, Jesus Marco, David Rodriguez, Witold Cieslikiewicz, Barbara Paplinska



CONTENTS

1. Introduction	4
1.1. Purpose	4
1.2. SCOPE	4
1.3. References	6
1.4. Overview	6
2. Overall description	7
2.1. Product perspective	7
2.1.1. System interfaces	12
2.1.2. User interfaces	12
2.1.3. Hardware interfaces	12
2.1.4. Software interfaces	12
2.1.5. Communications interfaces.....	13
2.1.6. Memory constraints	13
2.1.7. Operations.....	13
2.1.8. Site adaptation requirements.....	15
2.2. Product functions	15
2.3. User characteristics	16
2.4. Constraints	16
2.5. Assumptions and dependencies	17
2.6. Apportioning of requirements	17
3. Specific requirements	18
3.1. External interfaces	18
3.2. Functions	18
3.3. Performance requirements	19
3.4. Logical database requirements	19
3.5. Design constraints	19
3.6. Standards compliance	19
3.7. Software system attributes	19
4. Appendixes	20
5. Index	24

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to define the requirements and use cases for meteorological application. This application in the CrossGrid project includes three different subtasks. The first one, (1.4.a) addressing sea wave modelling forced by the winds simulated from a Coupled Ocean/Atmosphere Mesoscale Prediction System [5]. The second one (1.4.b) covers Data Mining techniques on Distributed Databases in the Grid environment applied to downscaling weather forecasts, and the third one (1.4.c), is focused on an implementation of air-pollution models. This is data-intensive, high-performance computing application, which require the efficient management and transfer of gigabytes of information in wide-area, distributed computing environment. The task includes both data mining and high computing applications. A large community of researchers shares massive data sets. These researchers need efficient transfer of large data sets to perform analyses in their local sites. In many cases different participating institutions create local copies and replicas of the simulated and observed data sets to overcome long wide-area data transfer latencies. The data management environment should provide not only security services such as authentication of users and control over who is allowed to access the data, but must provide tools able to determine where all existing copies are located to give the user possibility to decide whether to access an existing copy or create a new one to meet performance needs of their applications.

1.2. SCOPE

The weather forecast and wave modelling application will use the COAMPS [7] atmospheric model to simulate the nested spectrum of weather conditions over the Baltic Sea and to produce initial and boundary data for wave models. The application will use extensively the WAM4 [8] and SWAN [1] wave models for detailed estimation of sea conditions in selected cases. The atmospheric part of weather forecast application is most demanding for network resources. This application needs efficient resource manager, able to estimate the performance of the task, identify suitable alternative resources and to migrate the computation to another site when the actual performance is too low.

The meteorological data mining applications will use global meteorological data bases created by reanalysis projects of NCEP/NCAR and ERA-15, the mesoscale data archive generated by the operational UMPL system and local databases with observations from Spanish and Polish meteorological stations and posts. Data mining applications will be based on machine independent, structured data format. During the project at least two data mining algorithms will be developed and implemented to selected cases. An scheme for distributed unsupervised learning data mining technique will be implemented to global reanalysis data to produce realistic prototype for weather forecasting application using SOM (self-organising maps). Mesoscale data archive will be used to develop the methods able to improve direct atmospheric model output. We planned to use B-splines smooth of data with standard error bands and bootstrap technique to complete implementation of nonparametric or parametric maximum likelihood model inference. Typical task of the Markov Chain Monte Carlo (MCMC) method we planned to use is to draw a sample from joint distribution of random variables. This is done by Gibbs sampling, which produces a Markov chain whose stationary distribution is the requested joint distribution.



The air pollution application will use the STEM-II model [2] to simulate the behaviour near **thermal** power plants. It depends on the weather forecast data and it includes many chemical reactions and transport simulations for both dry and wet conditions. We will start using data for the As Pontes power plant as we have a lot of information about it, but the final product can be adapted to other environments. The parallelization of this model will provide real time reactions in the power plant. To develop the codes we only need the standard MPI library and to execute it, transparent access to the nodes on the GRID.

Definitions, acronyms, and abbreviations:

API	Application Programming Interface
COAMPS	Coupled Ocean-Atmosphere Mesoscale Prediction System
CrossGrid	The EU CrossGrid Project IST-2001-32243
DataGrid	The EU DataGrid Project IST-2000-25182
DBMS	Database Management System
DFS	Distributed File System
EDG	The EU DataGrid Project IST-2000-25182
GUI	Graphical User Interface
GRADS	The Grid Analysis and Display System (visualization package)
HDF5	Hierarchical Data Format version 5
HTTP	Hypertext Transport Protocol
JDL	Job Description Language
MPI	Message Passing Interface
MPICH-G2	Grid-enabled implementation of MPI
NetCDF	Network Common Data Format
O/R DBMS	Object/Relational DBMS
UMPL	Unified Model for Poland area - operational atmospheric model running at ICM
WAM4	Wave Model Cycle 4
SHTTP	Secure HTTP
SQL	Structured query language
SOAP	Simple Object Access Protocol
SOM	Self Organizing Maps
STEM-II	Sulfur Transport Eulerian Model version 2
SSL	Socket Security Layer
SWAN	Simulating Waves Nearshore model
VIS5D	Visualization Package from Wisconsin University
WSDL	Web Service Definition Language
XML	Extended markup Language



1.3. REFERENCES

- [1] Booij, N., R.C. Ris & L.H. Holthuijsen, 1999: A third-generation wave model for coastal regions. I: Model description and validation. J. Geophys. Res., 104, No C4, 7649-7666.
- [2] Carmichael, G.R. et al. (1991) The STEM-II regional scale acid deposition and photochemical oxidant model. Atmospheric Environment, 25A, 10, 2077-2090.
- [3] Fahrmeir L. & G. Tutz, 2001: Multivariate Statistical Modelling based on Generalized Linear Models. Springer Series in Statistics, 518 pp.
- [4] Hastie T., R.Tibishirani & J.Friedman, 2001: The Elements of Statistical Learning. Springer Series in Statistics, 533 pp.
- [5] Hodur R.M., 1997: The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS), Mon. Wea. Rev., 125, 1414-1430.
- [6] Michelson D., 2001: Use of the HDF5 file format for storing BALTEX Radar Data Centre sets, SMHI, 16 pp.
- [7] Mirin A.A., G.A. Sugiyama, S. Chen, R.M. Hodur, T.R. Holt & J.M. Schmidt, 2001: Development and performance of a scalable version of a nonhydrostatic atmospheric model.
- [8] WAMDI Group, 1998: The WAM model - A third Generation Ocean Wave Prediction Model. J. Phys. Oceanogr., 18, 1775-1810.

1.4. OVERVIEW

This document provides the software requirements for meteorological applications and proposes a number of selected cases for the wave modelling based on nested weather forecasts, for meteorological data mining as well as for air pollution applications.

Section 2 provides the overall description.

Section 3 provides requirements and relations with other WPs.



2. OVERALL DESCRIPTION

2.1. PRODUCT PERSPECTIVE

Daily output from atmospheric weather prediction simulations is comparatively large (in order of gigabytes). Data are not maintained in data bases but rather as flat files, typically stored in binary or structured data format such as NetCDF or HDF5 with associated metadata. Files are not updated once released. Researchers in the atmosphere and ocean modelling community will find it convenient to store local copies of portions of the data set. Therefore, the application has needs for managing copies of data sets at multiple locations, as well as higher-level services such as replica selection or automatic replica creation.

Databases in Meteorology provide the basic input for any analysis. Local observatories placed covering different regions of a country (see figure 1 for a Spanish example), provide simple observation databases including measurement of typical variables, like the temperature (T), geopotential (Z), wind field (U,V), humidity (H) in a given observatory placed at a certain length, latitude and height at a time. Measurements are provided typically every 6 hours or less.

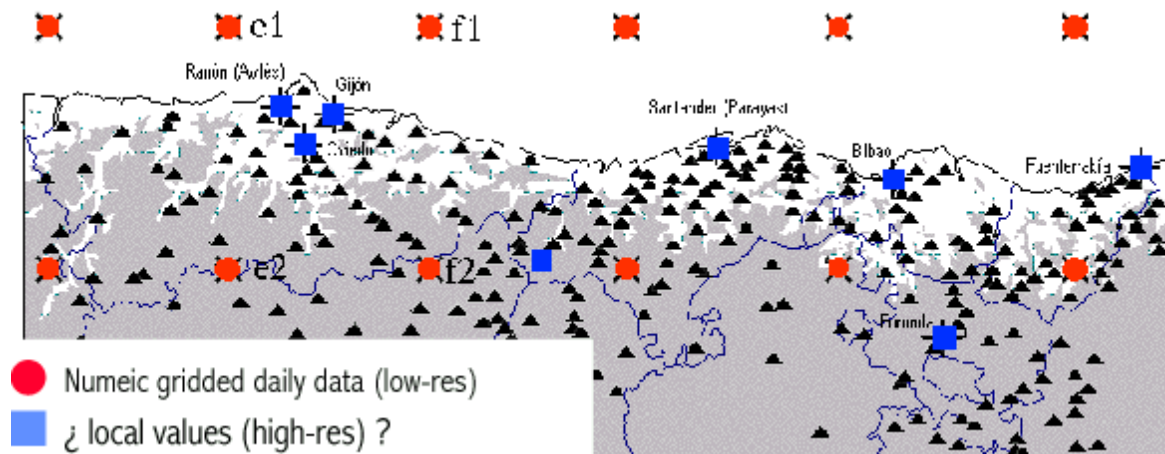


Fig. 1. Distribution of global model gridpoints and local observation sites used in data mining application

Reanalysis Models of Atmospheric Circulation provide global databases with information on a global grid covering the whole globe. The NCEP/NCAR reanalysis public database or the ERA-15 (restricted use) one, are two examples of such databases providing meteorological information in the form of (T,Z,U,V,H) tuples at different pressure levels (14-32) covering the whole globe with a 1x1 degrees grid (359x180). For use in local short-term prediction the relevant zone ranges from a 15x9 grid for a country (i.e. Spain) to a 60x40 grid for whole Europe. Typical sizes for this databases expanding over 15 years are O(GB). New reanalysis databases (ERA-40, DEMETER), with more detailed information over a longer period, will increase the typical size of the information to be analyzed to the O(TB) range.

Mesoscale database is stored in hierarchical directories on magnetic tapes and includes a 5 years long archive of every day data. In each day 8 assimilation cycles consist analyses of actual state of the atmosphere and 3h forecasts for Central Europe region with a spatial resolution of 17 km on 15 standard pressure levels are stored. Two times per day (at 00 and 12 UTC) the results of basic 48h forecast are stored.

SOM IMPLEMENTATION

Self Organizing Maps, an “Unsupervised Learning Technique” [4], constitutes a basic data mining tool very useful for large databases with large number of variables. It converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. As it thereby compresses information while preserving the most topological and metric relationships of the primary data items on display, it also be thought to produce some kind of abstractions.

The SOM usually consists of two-dimensional nodes on regular grid. A model of some observation is associated with each SOM node. The SOM algorithm computes the model so that they optimally describe the observations. The models are organized into a meaningful two-dimensional order in which models are closer to each other in this grid than the more dissimilar ones. In this sense the SOM is a similarity graph, and a clustering diagram too. It is a non-parametric, recursive regression process.

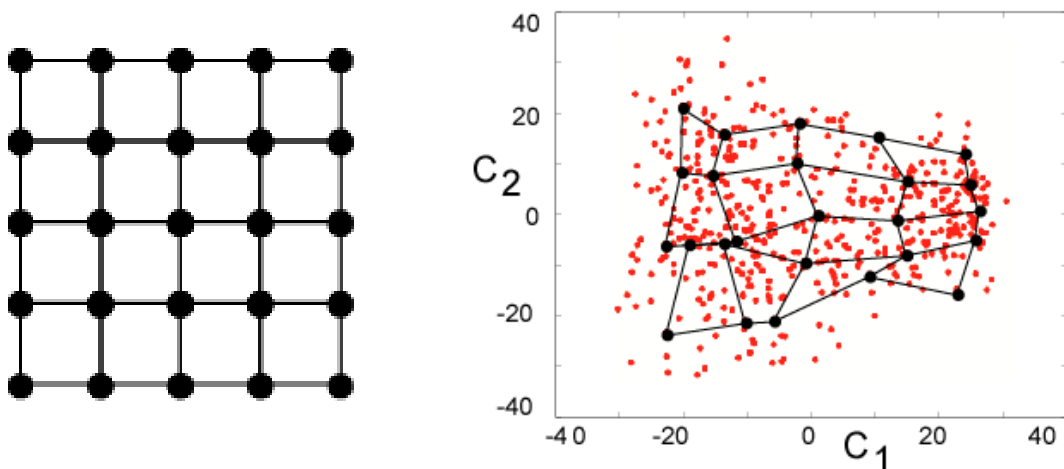


Fig. 2. SOM nodes and distribution of model vectors

The “on-line” version applies the regression of an ordered set of model vectors (arranged in the SOM two-dimensional grid) m_i in R^n , into the space of observation vectors $\{x\}$ in R^n using the formula:

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t))$$

where t is the sample index of the regression step, the regression is performed recursively, and $h_{c(x),i}$ is the so-called neighborhood function, with the index c (“winner”) defined by the condition:



$$\|x(t) - m_c(t)\| < \|x(t) - m_i(t)\| \text{ for all } t$$

This incremental regression process can be replaced by a “batch” version, following this scheme:

1. Initialize the model vectors m_i
2. For each one, index i , collect the list of all those observation samples $\{x\}$ whose model vector belongs to the neighborhood set of the node i .
3. Compute the new model vector as the mean over the respective list.
4. Repeat steps 2 and 3 until “convergence” of the procedure

Implementation of the SOM algorithm in a distributed way is based in the adequate partitioning of the database including the observation samples. As the total mean for each model vector is easily calculated from partial means for that same vector computed on distributed databases, the exchange of information is reduced to the model vector parameters updates and these partial means, while the large part of the information is not transmitted. Moreover, to cope with possible network effects in a grid configuration, the asynchronous update mode (where the MASTER waits for all WORKER nodes to update the info on the model vectors) can be relaxed to use a mode where the MASTER waiting time optimizes the convergence time, and new model vectors are updated waiting only to part of the total information. Obviously this requires a complete stripping of the databases information to avoid any unbalance that could affect the final results, a requirement already present in task 1.3, and a monitoring system (task 3.3) providing the necessary info to optimize this waiting process taking into account both the network and worker nodes.

To provide some typical numbers, we can consider a database with data corresponding to the national Spanish geographical 15x9 grid at six different heights and five variables (T,Z,U,V,H) (so dimensionality is $15 \times 9 \times 6 \times 5 = 4050$) along each 6 hours for 15 years ($4 \times 365 \times 15 = 21900$ tuples) working on the simplest 5x5 nodes SOM grid (i.e., $m_i, i=1,25$). So $25 \times 4050 \sim 100K$ weights (400Kbytes) are updated in each convergence step, while the total database size is about 1 Gbyte.

This will provide the first example to be tried in the next months.

VISUALIZATION OF SOM RESULTS

SOM can be seen as a “data visualization tool”, and as such it requires a good graphical interface to profit from its advanced data representation features. Currently the “Vis5D” package is one of those being employed for these and similar purposes in the meteorological applications. Vis5D (see <http://www.ssec.wisc.edu/~billh/vis5d.html>...), an open-source project, allows the user a simple 3D visualization, and the possibility to perform the required projections.

AIR POLLUTION MODEL

The STEM-II model will be initially applied to the As Pontes **thermal** power plant, located in the NW of Spain. Dynamic meteorological conditions and different emissions scenarios will be considered. Figure 3 shows the simulated distribution of two pollutants generated on the power plant using the STEM-II.



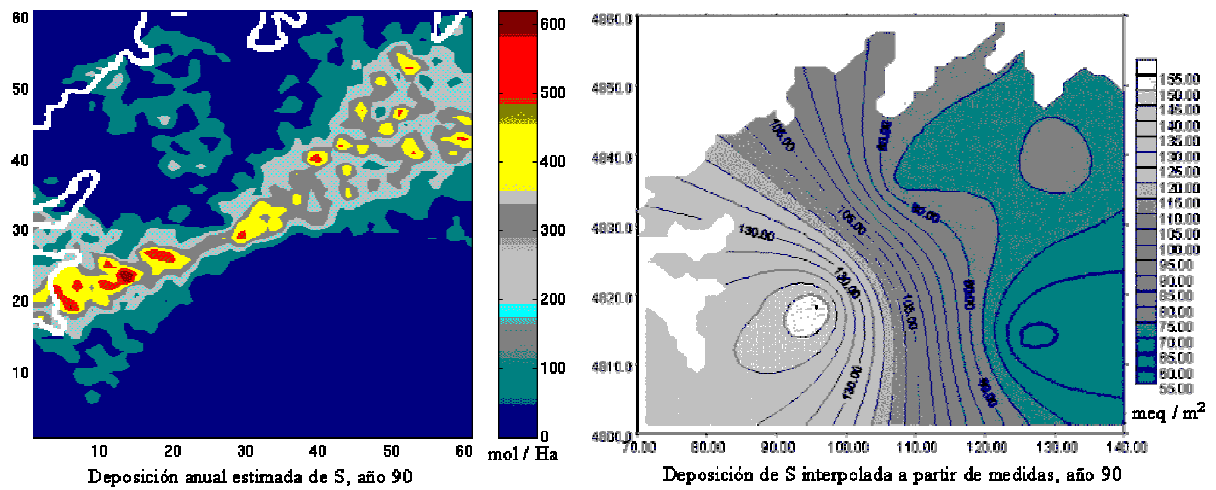


Figure 3: Simulation of the distribution of pollutants generated by the As Pontes power plant.

The structure of the STEM-II model is shown in the figures 4 and 5.

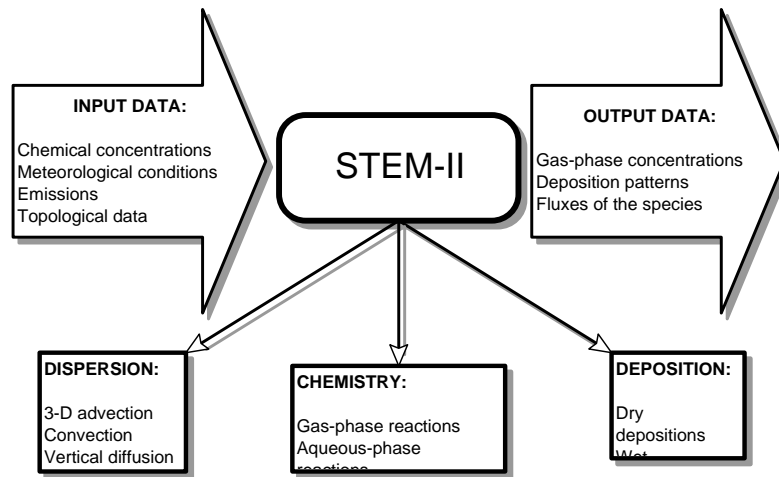


Figure 4. Structure of STEM-II



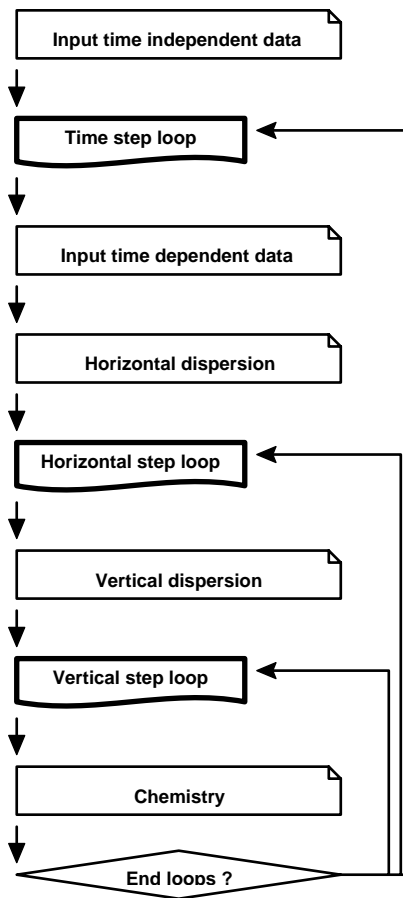


Figure 5: Pseudocode of STEM-II

The code needs stationary inputs: mesh characteristics, orography, etc, as well as time-dependent inputs from the emissions (basically amount of emitted pollutants) and weather forecast data from the meteorological application (humidity, wind speed, air density, etc) in each point of the mesh. This information must be provided by the weather forecast and updated in a file that is read by our program in each time step.

The code must be adapted to interact with the user, in such a way that he or she can directly change the weather or emissions conditions to check the behaviour under new virtual scenarios. In addition, one of the objectives of the final product is the fault-tolerance. We plan to provide the code with capabilities to check the fault in some node, and to recover the simulation from this point.

The visualization requirements for the air pollution applications are not critical, we only need to visualize the outputs in 2D and 3D diagrams showing the concentrations of pollutants in the simulation mesh or in the ground-surface. Most of the visualization tools

used by other applications (for example the one used in task 1.1) probably can be used by ours.

2.1.1. System interfaces

Application requires a secure, efficient data transport mechanism between storage systems and access to comparatively large amount of data by geographically distributed users for analysis and visualisation of the data. We need something similar to DFS (distributed file system) supported high-volume usage, data replication and local caching. We require also to have an access to structured data from a variety of underlying storage systems, exploring for example HDF5 facilities [6].

A universal grid data transfer and access protocol called GridFTP will provide secure, efficient data movement in Grid environment. This will be advantageous both to grid storage provider (ICM) and end users (maritime institutions).

Tools for authentication of users and control over who is allowed to access the data are also requested.

2.1.2. User interfaces

GUI module will be a new window splitted onto few parts:

- Atmospheric model desktop will have such options as domain of outermost model grid, resolution of the model, number of nested grids, duration of the forecast, the source of initial and lateral boundary conditions (i.e. from global NOGAPS or mesoscale UMPL models), preferred mode of computations (high performance computer or linux cluster). The tool for monitoring atmospheric model runs should have inform a new user on actually running tasks showing selected domains and lengths of forecast. The user can fit his requirements to existing run and get the results quicker than starting from begin.
- WAM4 model desktop will have such options as domain and resolution of the model, frequency of the storage of output data
- SWAN model desktop will have such options as domain and resolution of the model, frequency of the storage of output data
- The meteorological database desktop will have such choices as selection of meteorological elements, start date of the sample, a length of sample, number and position of grid points, selected method of data mining. User should have a possibility interactively use the graphical tools for analysis of results.
- The air pollution desktop must show simulated results for the concentration of each pollutant on each point of the simulation mesh. Some alarm signals have to be included in the GUI capabilities.
- The final product should allow the user to check the air pollution results, not only using real weather forecasts, but also under different hypothetical scenarios (in the weather or in the emission of pollutants conditions) that he or she can establish.

2.1.3. Hardware interfaces

No special interfaces with hardware components are considered at this moment.

2.1.4. Software interfaces



Proposed technologies:

- Java
- HTML
- Servlet
- MPI

2.1.5. Communications interfaces

Proposed protocols:

- GridFTP
- SSL
- SHTTP

2.1.6. Memory constraints

It depends on the granularity of the simulation, and on the weather forecast data. For the air pollution model, a typical situation on a 60X60X15 mesh need about 200 Mb of data. 23 MB are generated each hour of simulated time. Input and output information are less than 10 MB. This situation is for the current version of the code that is being used in the power plant. The size of the mesh and the time step can be adapted to the power of the computer.

2.1.7. Operations

We expect to develop the meteorological application in an interactive framework based on middleware tools provided by WP2 and WP3 according the following scheme:

1. An interactive user interface (UI) based on a Portal (task 3.1). The portal should support XML flow for input/output, and include the following “panels”:
 - a. Distributed Resources Use/Monitoring
 - i. Advance reservation for processing/storage
 - ii. Monitoring of Grid Load
 - b. Data description:
 - i. Dataset selection
 - ii. Data dictionary
 - iii. Derived “procedures”
 - c. Job description (in JDL , script language or WSFL)
 - d. Local persistency (coordinated with support for roaming access)
 - e. Interactive control (at least the basic “kill” sequence)
 - f. Graphical Output (with support for Java graphics and/or SVG graphics)
 - g. Alphanumerical Output
2. Grid interactive resources setup, defined via a Resource Manager for the corresponding Virtual Organization, including :
 - a. An Interactive Resource Broker taking care of finding adequate resources as described below, to establish an interactive session: Advance Reservation and Storage capabilities via Scheduler Agents (task 3.2). The reservation takes place with the following sequence:



- i. First the DATASET to be processed is analyzed: logical database file names are derived and then the corresponding physical database file names are found in the catalog
 - ii. The Interactive Distributed Databases Reservation Service is invoked, trying to find as close as possible in terms of network costs, in particular latency, a MANAGER node, several WORKER nodes, and several INTERACTIVE DATABASE SERVER nodes. Note that WORKER nodes and DBSERVER nodes can fit the same physical machine.
 - iii. If no suitable configuration is found, including an estimation of past and future costs, the Database Replication Service is invoked, migrating database files to the selected WORKER nodes if they allow for this storage reservation.
- b. Interactive Connection Manager Node (MANAGER), where the “main” job is executed, keeping connection with the user via an adequate mechanism and renovating it via a “keep-alive” mechanism, and with “WORKERS” across the grid in communication via “MPI” calls (with correctness verified in task 2.2) . It distributes the info to the different Grid nodes via MPICH-G2, and process the answer to get final results, that are then transmitted back to the UI.
 - c. A collection of computing nodes distributed across the grid testbed, WORKERS, processing the info according to the MANAGER instructions, and returning the output information to it.
3. A Testbed setup (defined in WP4), with a Virtual Organization (VO) dedicated to weather forecasting (CROSSGRID-WEATHER), including the possibility of interactive use of at least 100 WORKER nodes, distributed across at least three different sites, with replication facilities for databases.

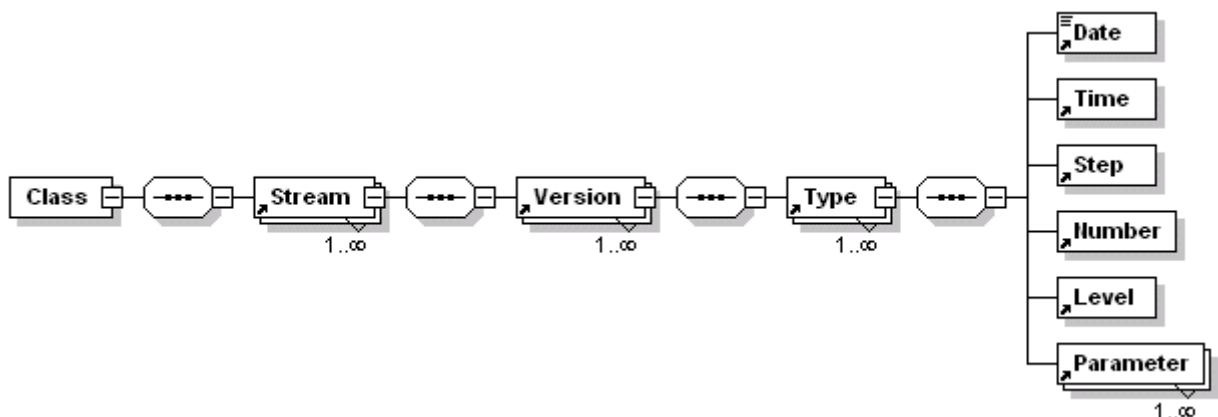
SETUP OF METEO DATABASES

The meteorological databases involved in this task are of three different types, local, mesoscale and global, but do involve similar information.

The first step so is the definition of an XML Schema for these data patterns. An XML instance is described below for better understanding.

```
<?xml version="1.0" encoding="UTF-8"?><Class xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="meteo.xsd"> <Stream model="Wave model"> <Version value="4096">
  <Type type="Forecast"> <Date>1967-08-13</Date> <Time>12:00</Time> <Step>24</Step>
  <Number>0</Number> <Level type="Pressure level">1000</Level> <Parameter
  table="ECMWF">Z</Parameter> </Type> </Version> </Stream></Class>
```

The corresponding diagram as provided by the XML-Spy tool is shown below.



This information is usually read from external formatted files, and will be stored in an O/R database. The XML schema is used in the database schema definition. This should take into account for performance the expected queries that will basically request the whole set of variables for a given geographical grid node.

The databases will be grid distributed and only very loosely coupled through a metacatalog. In practice it means that a "global database service" will be defined via this catalog and implemented via the corresponding "subdatabase services" implemented locally on a grid node. These grid nodes will run the database server, and provide the information in a three-tier scheme: a servlet will return in XML format the information to the client after querying the O/R DBMS database motor. The first implementation will use IBM IDS9.20, and implement queries via JDBC calls. The ResultSet, as indicated before, will provide an XML ResultSet output in the specified format. This scheme will be implemented in three steps: first in a local node (currently being done), then in a local cluster, and finally in the grid environment across at least three different testbed sites.

Adequate load balance is provided via database partitioning, and so it is an essential component. Also an adequate stripping of these databases is mandatory to avoid both loosening this load balance after any prefiltering, and possible effects due to partial processing of the database leading to possible systematic shifts in the results. *These requirements as other expected from other WP are those described in more detail in task 1.3.1.*

2.1.8. Site adaptation requirements

During a first phase of CROSSGRID (to the end of year 2002) atmospheric model will be executed in shared memory version, so first tests will be done at ICM local site. The results of these experiments will create the benchmark to distributed memory version on Linux cluster, which we planned to use in next years of the projects. The implementation of the air pollution simulation will be done exclusively using the message passing paradigm.

2.2. PRODUCT FUNCTIONS

The meteorological application will provide the following main features to end users:

- i) Functions connected to weather and sea wave forecasting
 - Access to the atmospheric, weather prediction model with services involving numerical simulation of the evolution of the atmospheric state in different, nested resolutions and geographical domains.
 - Access to the different sea wave models (WAM4, SWAN) with services allowing users to run these models for different scenarios depending on sets of initial and boundary data generated by the atmospheric model.
 - Access to visualisation tools (GRADS and VIS5D) allowing users to control sea wave and atmospheric model output in static and dynamic (animated) fashion.



- ii) Functions connected to data mining
 - Access to distributed meteorological databases with services allowing the user to choose selected subset of data available
 - Access to statistical packages performed specific nonlinear data mining techniques on modeled and observed data archives
 - Access to visualisation tools allowing the users to estimate results of data mining algorithms

- iii) Functions connected to air pollution simulations
 - Access to the air pollution sulfur transport model (STEM II) with services allowing the user to select and run air pollution model on distributed linux cluster.
 - Services which allow the user to change the set of parametrs driving the model, depending on the results of simulations, such as results about the chemical reactions and transport in the behaviour of the power plant. This information should be available each period of time (established by the user).
 - Access to visualisation tools allowing the users to estimate results of air-pollution simulations under different real or hypothetic conditions.

Functions common for all subtasks in meteorological application should allow users to check the status of their own pending jobs; to use results from similar, earlier run of another user; and to retrieve job output.

2.3. USER CHARACTERISTICS

The main users of applications and techniques developed in this task will be meteorologists, oceanographers, and air-pollution researchers. As an example, the group working in Santander under agreement with the Instituto Nacional de Meteorologia (INM, the Spanish National Institution for Meteorology) includes personnel from the Departamento de Matematica Aplicada y Ciencias de la Computacion de la Universidad de Cantabria, from the Consejo Superior de Investigaciones Cientificas (CSIC) working in the Instituto de Fisica de Cantabria (IFCA), and from the INM itself. These teams (5-10 people) work in the improvement of weather forecasting techniques, but also take care of implementing and running these applications using real data from observatories and providing short and medium time predictions for the local and national range. They will benefit from the new perspectives opened by the Grid framework, as their computational needs both in terms of computing power and distributed storage are highly demanding. The end-users of the air pollution model are the power plant engineers that have to decide actions according to the simulation output, and they can investigate the behaviour of the air pollution due to the plant under different situations.

2.4. CONSTRAINTS



-
- a) Hardware limitations (e.g., signal timing requirements) -TBD
 - b) Interfaces to other applications -TBD
 - c) Parallel operation: most of the computing modules of the application will run in parallel.
 - d) Audit functions -TBD
 - e) Control functions TBD
 - f) Higher-order language requirements: C, C++, Java, Fortran 77, Fortran 90
 - g) Signal handshake protocols: TBD
 - h) Reliability requirements: High.
 - i) Criticality of the application: TBD.
 - j) Safety and security considerations: Data privacy is a fundamental issue for Meteorological Institutions.

2.5. ASSUMPTIONS AND DEPENDENCIES

Supported platforms:

- Linux;

2.6. APPORTIONING OF REQUIREMENTS

Migrating Desktop from PSNC



3. SPECIFIC REQUIREMENTS

3.1. EXTERNAL INTERFACES

API to PHP and Java languages for efficient management of application

3.2. FUNCTIONS

The application shall:

- be platform independent;
- connect to the GRID Resource Manager to allow users to select resources and to submit their applications

Services involved:

Grid Portal Service (from task 3.1):

- UI input: message in XML including all sub-panel parameters plus keep-alive message
- Output to user: complete XML + XSLT transformation refilling pannels + update of “interactive” java applets for 3D graphics
- Persistence: saving to disk in XML format and in adequate local place + remote (for roaming use)

Monitoring Services (task 3.3)

- Input: Connection identifier defining allocated resources
- Output: status of network and Grid nodes

Resource Management Services (task 3.2)

- Advance Reservation Service for Interactive Use
- Scheduling Agents for Interactive Use

Interactive Distributed Database services (task 1.3.1)

- Location and best location service
- Stripping configuration service
- Replication & Installation service (task 1.3.1)
- Filtering/Permanent View service
- Simple Query service
- Special partial SOM processing service (DB side based)

Interactive Session Manager Services (task 1.3.1)

- Initialise session service
- Process User Request service: UI request (via XML) , Resource Manager answer
- Distributed processing service



SOM and other data mining service:

- Input: global or mesoscale databases + relevant variables + data mining architecture (SOM or MCMC)
- Output: results of data mining algorithm

Meteorological Downscaling Service:

- Input: results of data mining algorithm + local database
- Output: local weather predictions

3.3. PERFORMANCE REQUIREMENTS

Performance of linux cluster should be comparable to the performance of HPC.

3.4. LOGICAL DATABASE REQUIREMENTS

The O/R DBMS system will support the SQL92 and SQL99 standards. All information will be described via the corresponding XML Schema and matched to the DBMS via O/R views.

Basic info includes tuples of meteo information (T,Z,U,V,H) collected for different places, at different times, and for several altitudes. Reanalysis and actual data will share this same scheme. Read only access is needed by the user in this application.

The data catalog will follow the Globus and EDG middleware idea.

Stripping of databases, to support load balancing, will be implemented using directly different databases chunks completely decoupled.

3.5. DESIGN CONSTRAINTS

The application will run in Intel IA32 architecture, under the Linux operating system (versions 7.2 and above) and will be developed mainly in C++ and Java.

The SOM as data/mining procedure will be implemented in a Grid-friendly way, for example as a Grid Service. MPI calls will be used for the distribution across Grid nodes of the parallelized computing tasks.

3.6. STANDARDS COMPLIANCE

Information stored in databases will be described using the corresponding XML Schemas, and the corresponding DB Schemas implemented as O/R views.

3.7. SOFTWARE SYSTEM ATTRIBUTES

The most important attributes of the software are secure, reliable, efficient transfer of data in wide area environments and the ability to register and locate multiple copies of data sets. Maintainability, security and portability of the software are also desirable.



4. APPENDIXES

4.1 Use case for the weather prediction and sea wave modelling

Assumption: The routine mesoscale weather forecast for the next 48 hours indicated the strong wind conditions for Polish coastal zone of Baltic Sea. Researchers from three maritime institutions (IM, IH and IOUG) decide to investigate this situation in more detail for three different sub-bassins of the Baltic Sea. Working independently each other they decided to run triply nested atmospheric model to create the set of initial and boundary conditions for their wave models. Researchers from IM decided to apply WAM4 sea wave model to Gdansk Bay domain, researchers from IH want to apply WAM4 model to the Pomerania Bay and researchers from IOUG planned to use SWAN model in the Gdansk-Sopot-Gdynia coastal zone sector in western part of Gdansk Bay.

Step 1: Using atmospheric model desktop users from all three institutions selected the domain of the outermost model grid, number of nested grids and duration of the forecast. IM and IH chose the UMPL model as the source of initial and lateral boundary conditions, IOUG chose the global NOGAPS model for this purpose.

Step 2: The system estimated that the domain of the outermost grid is similar for all users. The system found that boundary and initial conditions from UMPL model are available on ICM server and that boundary and initial conditions from NOGAPS model are available on the GODAE server. System estimated the average time of data transmission to the GRID nodes showing this information to the users.

Step 3: System requested 100 nodes for atmospheric model run for outermost grid defined by users.

Step 4: IOUG decided to change his request for boundary data from NOGAPS to UMPL model, as estimated time of the availability of the data from the second model is sufficiently shorter (more than 1 hour).

Step 5: Users get access to the results of simulations from the outermost atmospheric grid. They apply these results to their sea wave models using appropriate desktops and at the same time they requested the system to visualize the selected atmospheric fields.

Step 6: The application started three simulations by the atmospheric model for the second nested grid. Each user has been informed on estimated time of computations. IOUG user decided to change his second domain to the domain of IM user and to use his output as boundary conditions.

Step 7: Users get results from their sea wave models simulations with winds from the outermost atmospheric model grid. Users requested the visualisation server to animate the wave model results.

Step 8. Finally, the application attempts to run of an atmospheric model for three different innermost model grids. Users got an information on resources available on the GRID and on estimated time of getting the final results.

4.2 Use case for the data mining (SOM)



Assumption: User would like to downscale the global model results to the local mesonet network using data mining technique based on self organising maps method.

Step 1: Using data mining desktop user checks the content of the global data base and selects the meteorological elements he thinks are important for his statistical model. Such parameters as start date of the sample, the length of the sample, number and position of selected grid points are also provided by the user.

Step 2: Request of the user is completed on the global data base server. The system sent the message to the user program with information in which nodes the data are available.

Step 4: The user checks the content of the observation (local) data base and selects all parameters needed for extraction of the local information for his data mining algorithm.

Step 5: User starts data mining process using meteorological information from global and local databases and parameters specific to the implemented method.

Step 6: Meteorological Downscaling Service: input: SOM + local database, output: local weather predictions

4.3 Use case for the data mining (MCMC)

Assumption: User would like to improve direct mesoscale model output using as tools the Bayesian approach via Markov Chain Monte Carlo simulations for inference in generalized additive and varying coefficients models, including mixed models with structured or unstructured random effects.

Step 1: Using data mining desktop user checks the content of mesoscale data archive and selects the meteorological elements she thinks are important for her statistical model. Such parameters like a start date of the sample, the length of the sample, the number and the position of selected grid points are also provided by the user.

Step 2: If the sample requested by the user is available in any repository on the GRID, the system provides it for the user. If the sample is not available, the system contacts the server (node) which is responsible for the management of the mesoscale database and the server initiate the process of preparation of requested data from magnetic tape storage system. User is informed on estimated time of such operation.

Step 3: Request of the user is completed on the mesoscale data base server. The system sent the message to the user program with information in which node the data are available.

Step 4: The user checks the content of the observation (local) data base and selects all parameters needed for extraction of the local information for her data mining algorithm.

Step 5: User starts data mining process using meteorological information from mesoscale and local databases and parameters specific to the implemented method.

Step 6: User analyses the results of data mining via MCMC method looking on alphanumeric and graphical output.

Step 7: Based on results of first data mining process user changes the parameters of MCMC algorithm and restart the data mining processes in nodes available for this task.



4.4 Use case A for the air pollution model

Assumption: The system is used to check the predicted behaviour of acid deposition in the neighbourhood of the thermal power plant. The user is a power plant engineer.

Step 1: The user sent the program to execution on the GRID, specifying the granularity of the simulation mesh. He or she specifies the time step, and the time between updates of results. All these decisions have influence on the performance of the parallel code, so the user must know these constraints to decide the value of all these program parameters.

Step 2: The simulation starts and gets new meteorological data as soon as it is available. The outputs are shown to the user on the GUI. Alarm and warning signals are included in the GUI system.

Step 3: The user can change the execution parameters: granularity, time steps, etc as he or she wants faster answer or more precise results.

Step 4: If some constraints are not satisfied, the user can take some actions, like decreasing the production of the power plant, or change the kind of coal.

4.5 Use case B for the air pollution model

Assumption: The system is used to investigate the influence of new conditions in the prediction of the behaviour of acid deposition in the neighbourhood of a power plant. The user is a power plant engineer.

Step 1: The user sent the program to execution on the GRID, specifying the granularity of the simulation mesh. Using the GUI system, he or she can change the information about weather conditions, to see for example what if now the humidity increases and it is raining, or what if the wind is twice fast or if the emission of pollutants increase.

Step 2: The simulation starts and gets new results.

Step 3: The user can change the conditions of the simulation to check new scenarios.

Step 4: The user can learn a lot about the air pollution under different conditions and be prepared for them as soon as they in fact occur.

4.6 Use case C for the air pollution model

Assumption: The system can ask more accuracy when critical situations are predicted. The user is a power plant engineer.

Step 1: The user sent the program to execution on the GRID, specifying the granularity of the simulation mesh. He or she specifies the time step, and the time between updates of results.

Step 2: The simulation. Some alarm or warning is detected.



Step 3: The program can change the execution parameters: granularity, time steps, etc. to get more precise results for this critical situations. This requires more computational power from the GRID, and then, the program can ask for more computational power to the GRID.

5. INDEX

Atmospheric model	4,5,6,12,15,20
Air pollution	1,5,6,9,12,15,16,21,22
Interfaces	11,12,16,18
data base	4,7,21
Data mining	1,4,6,7,8,12,15,19,20,21
Portal	13,18
Use case	4,20,21,22
Weather forecast	1,4,5,12,16,20

