



DELIVERABLE D1.7

REPORT ON FINAL APPLICATION DEMONSTRATION

WP1

Document Filename:	CG1-D1.7-v5.0-UVA007- ReportFinalApplicationDemonstration.doc
Work package:	WP1 CrossGrid Application Development
Partner(s):	UvA, Univ. Linz, CSIC, ICM, II SAS, INS, U.S.C., CYFRONET, UAB
Lead Partner:	UvA
Config ID:	CG1-D1.7-v5.0-UVA007- ReportFinalApplicationDemonstration
Document classification:	PUBLIC

This document describes the demonstration to be shown during the final Review.

Delivery Slip

	Name	Partner	Date	Signature
From	Alfons Hoekstra	UvA	6/12/2004	
Verified by	Robert Pajak	CYFRONET	1/02/2005	
Approved by	Alfons Hoekstra	UvA	1/02/2005	

Document Log

Version	Date	Summary of changes	Author
1.0	06-12-2004	Template	Alfons Hoekstra
1.1	13-12-2004	Input from WP 1.1	Elena Zudilova, Paul Heinzlreiter
2.0	16-12-2004	Input from WP 1.2 & WP 1.3	Branislav Šimo, David Rodriguez Gonzalez
3.0	20-12-2004	Input from WP 1.4	Juliusz Gajewski
4.0	21-12-2004	Minor modifications for WP 1.1, WP 1.2 and WP 1.3	Elena Zudilova, Branislav Šimo, Juliusz Gajewski
5.0	01-02-2005	Improving the overall document according to comments of internal reviews	Elena Zudilova, Branislav Šimo, David Rodriguez Gonzalez, Juliusz Gajewski
		Verified by the Quality Engineer	Robert Pajak

CONTENT

1. EXECUTIVE SUMMARY	4
2. DEFINITIONS, ACRONYMS AND ABBREVIATION	6
3. DESCRIPTION OF THE FINAL APPLICATION DEMONSTRATION	8
TASK 1.1: INTERACTIVE SIMULATION AND VISUALISATION OF A BIOMEDICAL SYSTEM	8
TASK 1.2: FLOODING CRISIS TEAM SUPPORT	12
TASK 1.3: DISTRIBUTED DATA ANALYSIS IN HIGH ENERGY PHYSICS.....	17
TASK 1.4: WEATHER FORECAST AND AIR POLLUTION MODELLING	20

1. EXECUTIVE SUMMARY

Being almost at end of the project, the CrossGrid consortium met in Lisbon, Portugal in the last week of October 2004 for an integration meeting, where all CrossGrid developers had a final check of the integration of their software and made the last preparations to the final demonstration to be held in February 16, 2005 in Amsterdam in conjunction with the 1st European Grid Conference. This deliverable reports on the status of the CrossGrid applications with respect to the final demonstration.

During the final demonstration in February 2005 task 1.1 is going to demonstrate the Grid-based problem-solving environment for the vascular reconstruction procedure, which includes the improved version of the time-dependent LBM solver for blood flow simulation, a semi-immersive interaction-visualization front-end for the exploration of patient's conditions and applying the bypass procedure (VRE system) and the GVK (Grid Visualization Kernel) for the Grid-enabled visualization of the results of blood flow simulation. We have incorporated the VRE system into the Grid via the MD. We have implemented a series of Java portlets to monitor the simulation progress using the lightweight version of the CrossGrid portal on a PDA. We achieved the secure Grid access, node discovery and registration, Grid data transfer, application initialization, medical data segmentation, segmented data visualization, LB mesh creation, job submission, distributed blood flow visualization, and bypass creation. During the final demo we are going to demonstrate the virtual surgery procedure on the Grid, where the tablet PC and a PSS system will form the experimental interaction-visualization set-up for task 1.1.

The flood forecasting application (task 1.2) consists of several simulation models (meteorological, hydrological and hydraulic) and appropriate post-processing tools connected together, so constituting a workflow. The workflow can be processed by the grid resources with a single click by exploiting the workflow service which manages its whole execution. Results of the whole simulation and each of its steps can be registered to the replica manager and described by metadata stored in the metadata catalog service. The metadata catalog can be browsed and searched later on in order to retrieve the registered files. Abovementioned functionality can be accessed using two user interfaces: a web based application portal and the MD. They both provide workflow management features. Portal provides complete data management; the MD provides just metadata search and browsing. The flood portal portlets have also been incorporated into the OGCE collaborative environment giving user a possibility to work with simulations and collaborate with other users in the same environment.

Task 1.3 plans to show a demo consisting of two parts: the interactive 'histogramming' and the ANN training. Both will use the MD as user interface. The first part is prepared based on the sequential version of an application. Even though the parallel version is available as well, we will wait until the MD supports it. This is also the case for the interactive clean stop of the ANN training. The plug-in for presenting the graphics of the ANN in the MD has is not ready yet. The Sleuth application, which is the subtask of WP1.3, is finished and can be demonstrated depending on time availability.

Task 1.4 is related to meteorological modelling, atmospheric pollution and wave models, for which meteorological model has been used as a driving force data delivery service. The final presentation of the application will consist of a demonstration of use of the CrossGrid developed tools for submission of computational tasks, their monitoring and visualisation of their outputs. This includes Portal and MD interfaces for different models. The purpose of the show will be the demonstration of how the models have been ported to the Grid environment, how they have been integrated and what kind of Grid technologies have been used for the optimisation of the simulation run.

To summarize, the application development is mostly finished and the current activities are concentrated on fixing minor problems. All tasks are integrated within the CrossGrid testbed, including the software developed in work packages 2 and 3 (i.e., MD, GPM, OCMG, Gridbench,

Marmot, Production testbed), and are actively preparing to the final demo. More detailed information can be found in the recent WP1 publications, available at the CrossGrid dissemination webpages.

2. DEFINITIONS, ACRONYMS AND ABBREVIATION

Glossary:

ALADIN	Meteorological model
ALADIN/LACE	ALADIN model for Central Europe
ANFAS	data fusion for Flood Analysis and decision Support (IST-1999-11676)
ANN	Artificial Neural Network
MM5	Meteorological simulation model
BFGS	Broyden-Fletcher-Goldfarb-Shannon Hybrid linear method for NN's
CAVE	Cave Automatic Virtual Environment
CAVELib	API for the development of applications for immersive displays
CAVETalk	API for the speech recognition support for the CAVE environment
CE	Computing element (EDG)
CESGA	Centro de Supercomputación de Galicia
CHAGAL	Graphical tool designed for visualization of 2D meteorological fields
CrossGrid/CG	The EU CrossGrid Project IST-2001-32243
DaveF	Hydraulic model
Desktop-VRE	Desktop version of the VRE system
DataGrid/EDG	The EU DataGrid Project IST-2000-25182
DBMS	Database Management System
DRIVE	Distributed Real-time Interactive Virtual Environment
Flood-VO	Flood Virtual Organization
G-PM	Grid Performance Monitoring tool
GRID	Grid framework for sharing of distributed resources
GUI	Graphical User Interface
GVK	Grid Visualization Kernel
HLA	High Level Architecture
GVK-BFVS	Grid Visualization Kernel Blood flow Visualization Service
HSPF	Hydrological model
JDL	Job Description Language
LBM	Lattice Boltzman Method
LCG	Large Hadron Collider Computing Grid
MARMOT	MPI verification tool
MCMC	Markov Chain Monte Carlo (statistical inference method)
MD	Migrating Desktop
MLP	Multi Layer Perceptron
MM5	Meteorological simulation model
mlpfit	A tool for Multi layer Perceptrons
MPI	Message Passing Interface
MPI-G2	Message Passing Interface with the globus2 device
MPICH-G2	Grid-enabled implementation of MPI
NLC	Rainfall-runoff hydrological simulation model
OCM-G	Grid OMIS (On-line Monitoring Interface Specification) compliant monitor
OpenGL	Low-level graphics library specification
OGCE	Open Grid Computing Environments Collaboratory
OGSI	Open Grid Services Infrastructure
O/R DBMS	Object/Relational DBMS
PSE	Problem Solving Environment
PSS	Personal Space Station

PAW	Physics Analysis Workstation software
R	Public domain statistical package
RAS	Roaming Access Server
RB	Resource Broker
ROOT	An Object-Oriented Data Analysis Framework
QA	Quality Assurance
SE	Storage Element (EDG)
SHMI	Slovak HydroMeteorological Institute (Subcontractor of II SAS)
SM	Standard Model (in the High Energy Physics context)
SOM	Self Organizing Maps (statistical clustering method)
STEM-II	Sulphur Transport Eulerian Model – II
TIN	Triangle Irregular Network
UI	User interface
UCI	file format (User's Control Input)
VRE	Virtual Radiology Explorer
VR	Virtual Reality
VRVS	Virtual Rooms Video Conferencing System
VTK	Visualization Toolkit
WMO	World Meteorological Organization
WN	Working Node
WS	Web services
XML	Extended Markup Language

3. DESCRIPTION OF THE FINAL APPLICATION DEMONSTRATION

TASK 1.1: INTERACTIVE SIMULATION AND VISUALISATION OF A BIOMEDICAL SYSTEM

Task leader: Elena Zudilova, UvA (CR5)

During the Review in February 2005 in Amsterdam we are going to demonstrate the final prototype of the biomedical application. This prototype is a Grid-based problem-solving environment (PSE) for virtual vascular surgery. We use a set of hardware and software resources available via the CrossGrid infrastructure for building a specific framework to support vascular surgeons and interventional radiologists in their pre-operative decision-making. With such a PSE, medical experts will be able to prepare for certain interventions in the vascular system of a patient, using a combination of advanced simulation and visualization tools.

Our final prototype includes the improved version of the time-dependent LBM solver for blood flow simulation, a VR-based interaction-visualization front-end for the exploration of patient's conditions and applying the bypass procedure (the VRE system) and the GVK (Grid Visualization Kernel) for the Grid-enabled visualization of the results of blood flow simulation.

We have incorporated the VRE system into the Grid via the MD Grid portal (Fig. 1-a). We have also implemented a series of Java portlets to monitor the simulation progress using the lightweight version of the CrossGrid portal on a PDA (Fig. 1-b). We achieved secure Grid access, node discovery and registration, Grid data transfer, application initialization, medical data segmentation, segmented data visualization, LB mesh creation, job submission, distributed blood flow visualization, and bypass creation. The VRE runs on a local machine, but it is launched and initialized through the MD. The input for the VRE is the segmented medical data produced at the Leiden Medical Center (LUMC). The CrossGrid testbed provides access to the medical data of interest from a medical image repository acting as a Grid SE in Leiden.

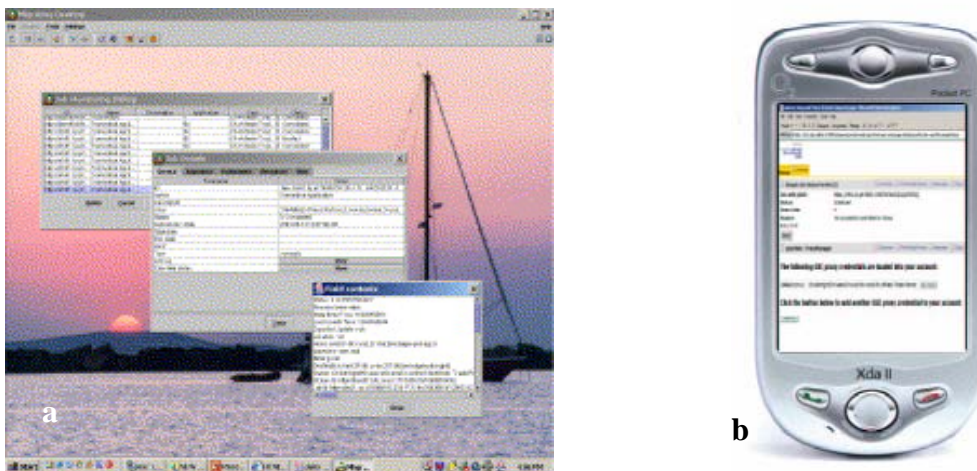


Figure 1: Simulation monitoring via the CrossGrid MD portal (a) and via the CrossGrid lightweight portal on a PDA (b)

The patient's blood flow is simulated using Grid resources. An efficient mesoscopic computational haemodynamics solver for blood-flow simulations is based on parallel cellular automata. We are able

to simulate pulsatile Newtonian flow in a straight rigid 3D tube. To allow for the parallel execution, the simulation volume is divided into several sub-volumes, and each sub-volume is processed concurrently. We use a curved boundary condition for the wall and the technique based on reducing the Mach number for minimizing compressibility errors. An example of our recent results of blood flow simulation is shown in Fig. 2-a.

To convert the medical scans into computational meshes, raw scanned data is first segmented so that only the arterial structures of interest remain in the dataset. The segmented data is then converted into a computational mesh. The generation of an accurate computational mesh from segmented data, for the flow solver, is another part of the VRE functionality that has been improved significantly and tested using the desktop version of the VRE system (Desktop-VRE). In Fig. 2-b the GUI of the current version of the Desktop-VRE is shown.

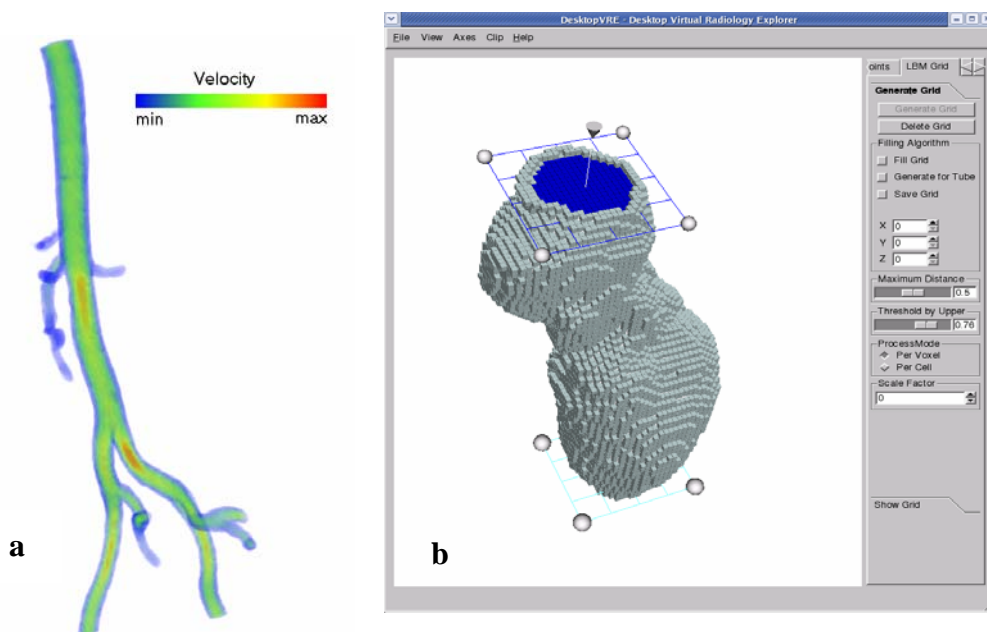


Figure 2:
a) Results of the LBM blood flow simulation for the case of the abdominal aorta
b) The generation of LBM mesh using Desktop-VRE

However, during the final demo we are going to show the VR-based version of the VRE provided with a multi-modal interface, which combines natural input modes of context sensitive interaction by voice, hand gestures and direct manipulation of virtual 3D objects. We call it the Virtual Operating Theatre, as a user can ‘play’ the role of a vascular surgeon planning and conducting the treatment of a vascular disease on a virtual simulated patient.

The GVK addresses the combination of Grid applications and corresponding visualization clients on the Grid. While Grids offer a means to process large amounts of data across distant resources, visualization aids in understanding the meaning of data. For this reason, the visualization capabilities of the GVK are implemented using Globus services, thereby providing Grid visualization services via dedicated interfaces and protocols while at the same time exploiting the performance of the Grid for visualization purposes. In addition to VTK-based glyph visualization module and the VTK-module for

isosurface extraction parallelized using MPI, the GVK also provides now support for the Grid-based pathline visualization of computed periodic blood flow patterns.

The task of the Grid Visualization Kernel Blood flow Visualization Service (GVK-BFVS) within the final demonstration of task 1.1 is the preparation of the blood flow solver's results for the final rendering step. It consists of two separate VTK-based visualization modules executed by a shellscript, which also handles the task of data preparation for the visualization task and the transport over the Grid.

When the GVK-BFVS service is invoked (e.g., from MD) it receives its input data in the form of a tar-archive as its standard input. This approach has been chosen in order to integrate the BFVS with the MD. After that the first visualization module gets invoked, which converts the renumbered velocity files into pathlines, which are stored as binary unstructured grid files. Some of the commandline parameters of the shellscript are used for executing the pathline conversion. Then the second visualization module is invoked, which converts the unstructured grids into binary VTK polydata frames. It also receives some of the parameters specified at the invocation of the shellscript. Finally, the VTK polydata frames are put into an archive file, which is written to the standard output where the MD can receive it.

For the final demonstration we plan to use an HP Tablet PC (Fig.3 right) for all 2D manipulations of the user such as: login to MD, job submission, monitoring, etc. and a PSS (Personal Space Station) as the 3D visualization-interaction front-end for the visualization of and interaction with the patient's data, obtained from a scanner, and local rendering of the results of the blood flow simulation after they have been processed by the GVK. All manipulations with a tablet PC and a PSS during the final demonstration will be shown enlarged on two big screens to the audience.



Figure 3: The experimental interaction-visualization set-up: a PSS (left) and HP Tablet PC (right)

A PSS (Fig.3 left) allows users to interact directly with a virtual world. A PSS consists of a semi-transparent mirror, in which a stereoscopic image is reflected. A user reaches under a mirror to interact with the virtual objects directly with his or her hands or by using task-specific input devices. However, there is a possibility to build a shared environment, where users can manipulate the same virtual objects working on different PSSs.

We are going to demonstrate the complete virtual surgical procedure on the Grid, which will include the manipulation with 3D images of the arteries generated from 2D scans transferred from the remote

image repository, adding a bypass, and changing manually its geometry and position. The blood flow in this new created geometry will be then computed using Grid resources to assess the effectiveness of the proposed treatment.

We conclude with the description of a preliminary scenario for the final demonstration of task 1.1:

1. A physician (or the user playing the role of a physician) starts the VRE system on a PSS.
2. In parallel, his assistant logs to the CrossGrid MD from a tablet PC using Grid certificate and private key.
3. The assistant checks if there are segmented or non-segmented medical data sets ready for analysis and simulation in one of the virtual nodes, and securely transfers a few in order to work with them locally.
4. When the data transfer is completed the physician loads the segmented medical data and starts manipulating with it within the VRE. He selects a region of interest, crops image, adds a bypass, and creates a LBM mesh.
5. After the LBM mesh is generated, within the portal, the assistant selects the biomedical application icon (parameters and files are taken from user's profile), and submits the job to the CrossGrid, to the geographically closest adequate CE in the Grid, using a replica manager service.
6. The assistant then checks job submission and/or simulation progress (via the portal) and demonstrates the results to the audience.
7. After the job has been completed, the velocities, pressure, and shear stress are transferred to the GVK for applying visualization services. The assistant demonstrates to the audience the progress with the GVK.
8. Finally, the visualization data processed by the GVK is transferred to a PSS, where it is rendered and reviewed by the physician.

At this moment the scenario implies downloading the portal, from Poznan Supercomputing and Networking Center, to a local roaming storage element (machine used as an access point); secure access to the testbed; virtual exploration of available SEs throughout the Grid; secure data transfer from an image repository SE at LUMC; the preparation of the data for the blood flow simulation within the VR-based VRE; job submission to the LBM solver via the RB, at Lisbon Instrumentation and Experimental Particle Physics Laboratory; and visualization of the simulation results using the GVK at the Johannes Kepler University Linz).

TASK 1.2: Flooding Crisis Team Support

Task leader: Ladislav Hluchy, II SAS (AC6)

The flood application tries to forecast a possible flood in the specific region and to compute borders of disaster areas by running several numerical models. Basic building stones are meteorological, hydrological and hydraulic simulation models created by third parties. They have been adapted for the grid environment by II SAS. II SAS has also created supporting grid services (workflow and metadata), a plug-in for Migrating Desktop application and portlets for application portal.

Figure 4 shows the architecture of the flood application. The application management core consists of two grid services – workflow engine and metadata catalog – that operate on other services provided by the CrossGrid and LCG grid middleware. These two services are accessed by two user interfaces – application portal and the MD. The application portal can be accessed by any graphical web browser and, therefore easily accessible from any operating system and environment. The MD requires a Java Virtual Machine installed on the client system, but its environment is more universal for working with grids.

The workflow engine uses the Job submission service (JSS) as a gateway for actual grid job submission and job state requests. JSS then uses the resource broker (RB) and logging & bookkeeping (LB) services to select the resources that will be used for job execution and for job state change events logging.

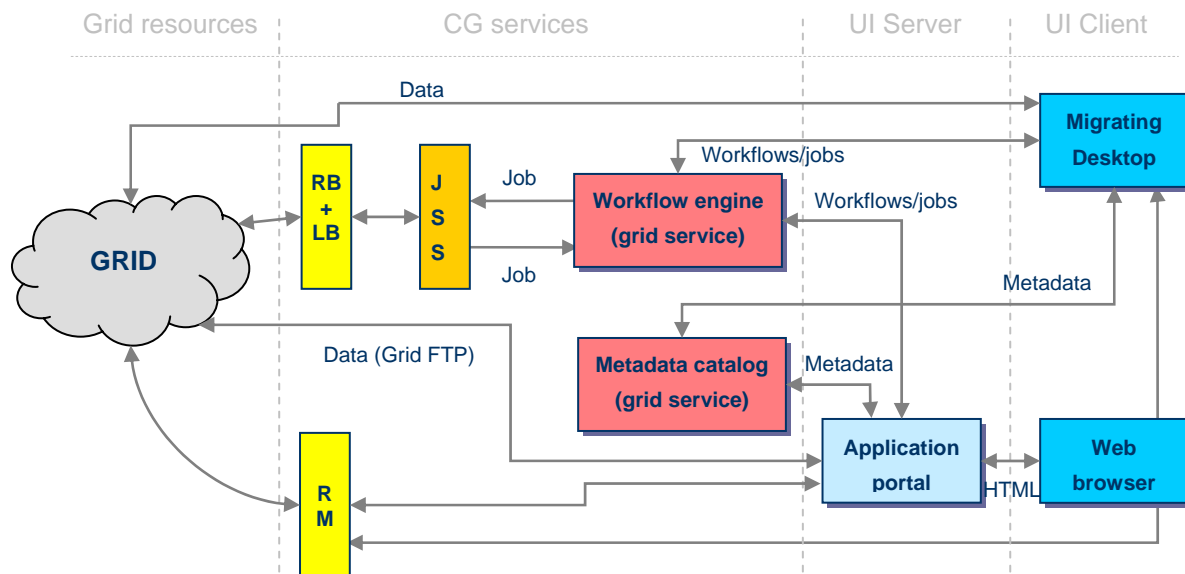


Figure 4: Flood application architecture

Metadata catalog stores the metadata about data files that are referenced using unique identifiers – GUIDs. A GUID is created for a file upon its registration to replica manager (RM) and can be later on used for its retrieval by RM.

Components not visible in the picture are the simulation models and post-processing tools that are hidden behind the workflow engine service.

Each of the components mentioned above will be described in more detail in the following sections.

Application Portal and Migrating Desktop

The application portal has been implemented on top of the Jetspeed portal framework that makes it possible to plug-in portlets – reusable portal components – in order to compose a user interface.

We have created following portlets:

- Workflow portlets – Workflow Template portlet allows a user to select one of the predefined workflow templates and instantiate it. The instances are shown in the Workflow portlet, which allows a user change parameters of jobs comprising the workflow, submit the workflow for execution, monitor its execution and also to browse and view the results of particular jobs in cooperation with visualization portlet.
- Metadata portlet – Metadata management tool that is capable to query, store and remove metadata describing specific file(s). It enables a user to register and un-register file replicas as well.
- Browser/Visualization portlet – Enables a user to view text and image files. It is used by the workflow portlet for job output browsing and visualization. It also allows a user to play a sequence of images as an animation (images have to be numbered sequentially).
- Edit portlet – Is used to edit outputs of a simulation if a user wants to run subsequent simulation with modified data.

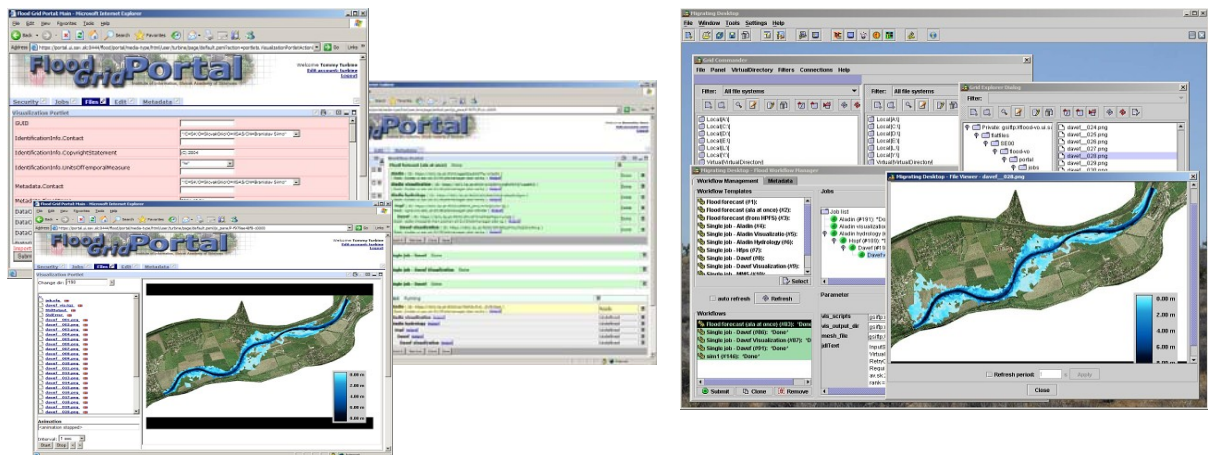


Figure 5: Application portal and Migrating Desktop running flood application

The Migrating Desktop is used in similar way using the flood application plug-in that enables a user to manage his workflows and browse the metadata database.

The advantage of using portal user interface is that user is required to have just the web browser for the operation. The MD requires a Java Virtual Machine be installed on the client side, but MD provides better user experience in terms of GUI response times, capabilities and extensibility.

Workflow Service

The workflow service is an interface to the workflow engine and database. It manages creation, execution and deletion of users' workflows. The main value of this service is its ability to run a whole workflow in the grid, handling the job dependencies automatically. The user just sets the parameters for all the jobs and submits the workflow for execution. Job or workflow state can be queried at any time.

It also serves as a repository of all workflows that have been run, are running or will be run, thus giving a user possibility to compare the results of various simulations.

Each workflow is an instance of one of the predefined workflow templates. Currently, there is no tool for workflow template creation and workflow templates have to be created manually in the database.

Workflow jobs are internally handled by a job type specific Java class. By implementing other job type classes the workflow can be enhanced to support various other task as part of the workflow, e.g., local tasks doing some post- or pre-processing or other grid-oriented tasks such as file staging and so on.

Metadata service

The metadata service allows users to add, remove, modify and find metadata, as well as files described by it. User is able to select files by stating (some of) their properties, to view properties of found files, to add new or to remove existing file descriptions from the metadata database.

Replica Manager (RM) is used to download a file found in the metadata catalogue. The RM can be accessed by the user via portal and migrating desktop, and by jobs running in the grid.

All generated files are automatically (as a final part of the simulation jobs) described by metadata and stored in a file. This metadata can then be used by the metadata service upon file registration. For this purposes a set of metadata generator jobs has been developed. These jobs can scan the generated files and construct metadata descriptions of them.

Simulation models

All simulation models and respective post-processing jobs are run in the grid in the same way; every job is represented by a wrapper script that accepts a configuration file containing job parameters. Only this wrapper script and the configuration file are part of the job submission request. When the script starts running on the grid resource, it will transfer all data and binaries needed for job execution. At the end, it will copy the output files to the specified storage element.

As the simulation models have been already described in detail in previous deliverables this chapter will just summarize the current state.

Two models are used for *meteorological* simulation – ALADIN and MM5. MM5 has been integrated into the cascade later during the project, but has the advantage of being freely available

Both meteorological models are parallel applications that use MPICH-P4 for interprocess communication. They were checked for MPI standard conformance using MARMOT tool and analyzed for performance bottlenecks using OCM-G and G-PM tools developed in CrossGrid. The analysis showed that the models are communication intensive (approximately 4 MB/s on 100Mbit ethernet network) and their scalability is up to 8-10 nodes depending on problem size. Thus, they are not suitable for distributing over more sites using MPICH-G2. Forecasts for 48 hours took about one hour depending on problem size. Because of uncertainty in weather forecast, different input data have to be processed and results compared and checked by experts for correctness. The models produce datasets for each forecasted hour that are processed by subsequent workflow module (post-processing). This decreases the time needed for producing near-future hydrological forecasts.

Input data for MM5 are being taken from Aladin boundary conditions for nearly whole Europe. Output data from MM5 are post-processed for visualization of pressure, precipitations, temperature and wind for various pressure levels. Second post-processing of output data generates 81 scenarios for hydrology (the aim is to minimize the error of localization of precipitations). To obtain better results from MM5 model, we need to integrate one-way and two-way nesting of MM5 model.

Main problems with meteorological models were to port the Fortran and C code to Linux and solve the compiler issues. Running of ALADIN model on Grid site requires agreement with responsible member of ALADIN consortium in the country where the site is located. Currently, it can run only in II SAS site. Therefore we are now focusing on MM5, which does not have such limitations.

There are two *hydrological* models used – HSPF and NLC. They both are sequential jobs with small execution times (up to seconds). Therefore no parallelization was needed. They profit from grid technology by multiple parallel executions with modified parameters. Such scenario is being implemented in our workflow system.

Both models use the same time-series input data files (generated by one of the meteorological models) and produce output files in the same format. Nonetheless, presence of HSPF code is needed to run NLC code, because HSPF tool is used for data conversion for NLC (it is invoked automatically by NLC script).

Configuration files for NLC and HSPF are excessively different, in quantity and quality, although they both specify terrain parameters. A user – specialist – should estimate these parameters. Default configurations were prepared for pilot site.

The *hydraulic* model used is DaveF, which receives terrain mesh and hydrographs as inputs and computes water level and velocity at each node of the mesh. Extensive hydraulic simulations for DaveF model calibration at Vah river pilot site have been carried out. A new mesh for whole pilot area has been created for the simulations and also a metadata generator for creating metadata from simulation output.

Flood visualization

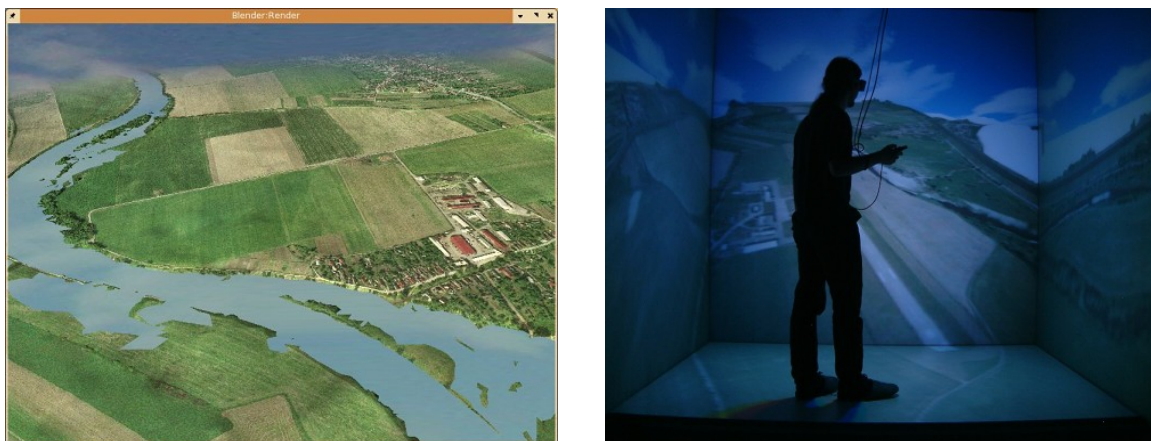


Figure 6: 3D flood visualizations: Blender output (left) and VRML in CAVE (right)

As a part of the project, three different solutions have been implemented for the flood visualization, i.e. the visualization of hydraulics simulation output:

- The first one is based on the GRASS GIS system. GRASS is a free, professional geographical information system offering a rich set of commands for work with raster, vector and site-oriented data. The implemented system produces simple and pertinent two-dimensional pictures, targeted mainly for professional users.
- The second solution uses Blender (free, open-source 3D modeling program) to create 3D pictures of the flooded area. Blender exposes a programming API, which made the automatization of many tasks possible. However, as Blender is an inherently interactive application, it always needs assistance from the user (adjusting camera, view, etc.), making the automatic workflow-driven processing impossible. It is a big disadvantage that made us move towards the VRML-based solution.
- The third, VRML-based solution, transforms the input data into the VRML language. VRML is an industrial standard for describing 3D scenes, so this solution is platform-independent. This

solution is written entirely in Python and requires no other software, what is the biggest advantage over the previous two solutions. The only requirement is a VRML browser, which has to be installed into the user's web browser.

The first, two-dimensional solution has been incorporated into the workflow as standard post-processing for the DaveF model. The third, VRML-based solution has not been incorporated into workflow yet, but its output has been successfully used in the CAVE to create a 3D virtual reality environment of the flooded area by our colleagues from the Johannes Kepler University in Linz.

Proposed presentation plan

We'd like to present the workflow submission using the application portal, showing the jobs starting to run on the resources in the testbed. Then results will be browsed, but due to the simulation taking about an hour to finish, results of already finished simulation will be shown. It will be also possible to see the results being produced by the running simulation.

During the result browsing a result set will be registered to metadata catalog with user filling in empty metadata fields in the registration form. Then it will be searched for specifying some of the metadata describing it and, finally, downloaded when found.

We are working on VRML integration into the workflow. With this task done, an interactive 3D visualization of the flooded area in the VRML browser will be presented.

All of the abovementioned actions will be run from the OGCE collaborative portal, into which the portlets have been incorporated.

TASK 1.3: DISTRIBUTED DATA ANALYSIS IN HIGH ENERGY PHYSICS

Task leader: David Rodriguez, CSIC (CR15)

Task 1.3 contains working prototypes of a parallelized Artificial Neural Network (ANN) and the Sleuth algorithm. It also contains development plugins to be used in the MD as well as distributed data access code (subtask 1.3.1)

We will concentrate on showing the interactive part of our work. This could imply two parts: the example histogramming application and our ANN application, both running from the MD. The reason of having these two parts is that the example histogramming application shows better the interactive possibilities of the CrossGrid project, but the ANN application along with Sleuth is the main software product of our task.

The histogramming application consists of two parts: a plugin for the MD, and a parallel (MPI) application. Once the interactive job is launched, the user must execute the plugin. This plugin contains a user interface consisting of three zones (see Fig. 7):

1. The *XML Histogram zone* is where the histograms are plotted (implemented using the JAIDA package).
2. The *Menu zone* has several buttons to control the program flow.
3. The *Application Parameters zone* is where the execution parameters should be assigned.

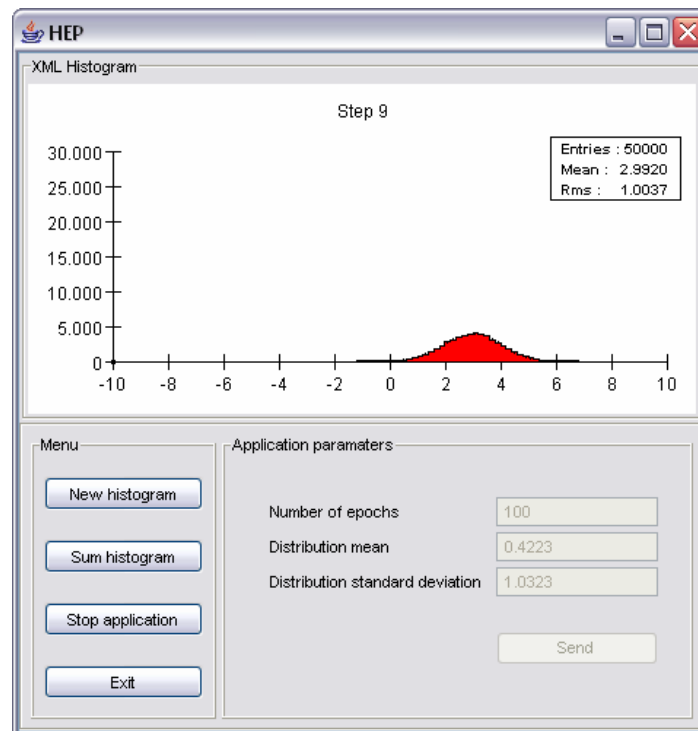


Figure 7: Histogramming application plugin

Once the execution has been started, the user has to choose a menu option by pressing the corresponding button. Usually it would start a New Histogram. Then the application parameters

should be filled in and sent. The application will start receiving the calculated distribution in XML format. This data is parsed and shown as a histogram in the upper zone.

Once this data is transferred completely, the user may want to create a new histogram from scratch, or to add more data to the current one. This iterative process goes on until the user presses the Stop Application button.

The ANN application is a parallel MPI application that trains a neural network in order to separate signal events from background ones. What will be shown is how to use it from the MD.

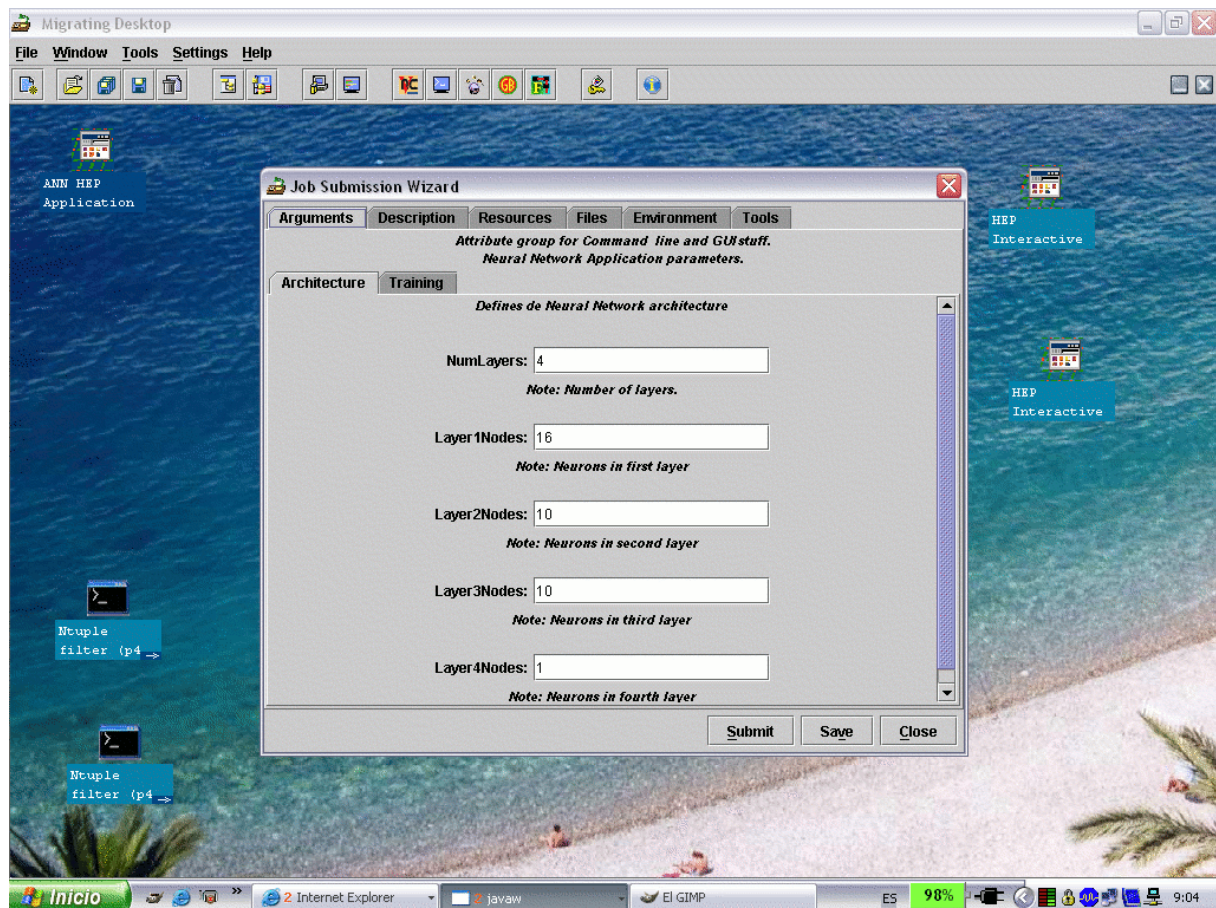


Figure 8: ANN Job Submission

We will show how the execution parameters are introduced and the input and output files set using the Job Submission Wizard. Then using a plugin we will monitor graphically the training progress.

This plugin shows a plot of the evolution of the error with the training epoch, the user can detect if there is any problem with the training (like overfitting) and decide to stop the job. A button will be provided for this purpose that could stop the job “nicely” using the interactive possibilities provided by the middleware and the MD. By now the only way to cancel the job is to kill it.

Apart from this interactive issue the main target of the demo will be showing the scalability of the ANN application. So we will execute the application with the same parameters and different number of working nodes to compare the speed of the training.

Sleuth algorithm functionality can be demonstrated using MD for the job submission, showing the progress of generating the Hypothetical Simulated Experiments (HSE) and for user's inspection of the

region of excess in distributions of physical variables. In addition, plots showing results of tests for scaling of performance with numbers of nodes can be provided.

TASK 1.4: WEATHER FORECAST AND AIR POLLUTION MODELLING

Task leader: Juliusz Gajewski, ICM (AC2)

During its demonstration task 1.4 will focus on:

- The presentation of the simulation submission for atmospheric forecast, including different scenarios of wave model simulations,
- The presentation of air pollution model and visualisation of atmospheric and air pollution model output.

The general data flow for applications developed in task 1.4 is shown in Fig. 9. The data flow in the task 1.4 applications is the natural organiser of the scheme of the presentation.

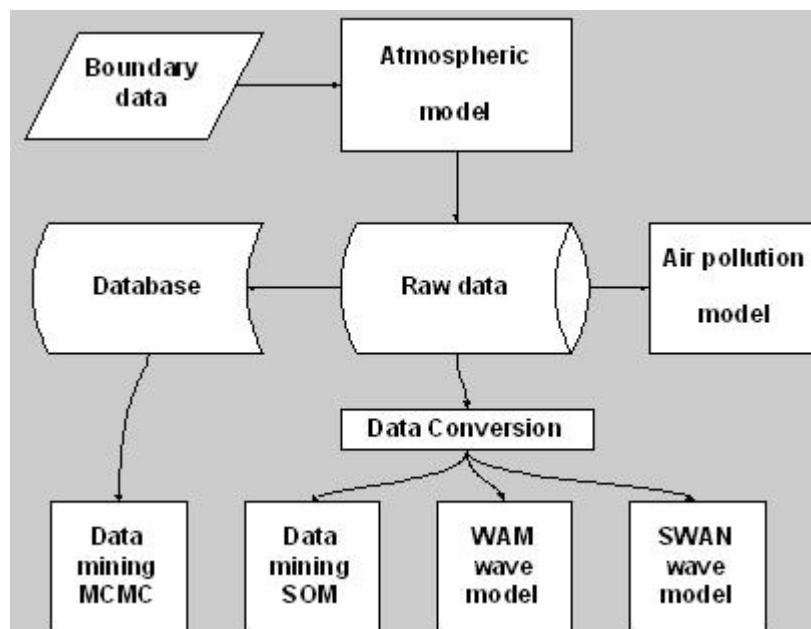


Figure 9: Scheme of data flow in applications of task 1.4

The atmospheric model COAMPS was implemented as main "production tool" with the purpose of generating initial and boundary conditions for selected applications. The model was designed in such a manner that is able to produce data for air pollution model STEM II, wave sea models WAM and SWAN, and data mining application SOM. The model set-up in CrossGrid consists of two parts – graphical user interface implemented using the MD and the MPICH-P4 application.

The separate scripts for each application are invoked from the Migrating Desktop GUI (Fig. 10) according the user choice. All needed initial data and results are stored Storage Elements. For the air pollution application, numerical data and visualised wind fields are stored in one-hour intervals; that gives the possibility to run the atmospheric model and the air pollution model in parallel.

The atmospheric model run is a heavy and time consuming task, the scalability of the model depends mainly on communication issues. When run in heterogeneous X# environment it is much slower than in specially tuned cluster or supercomputer. The main message from this X# implementation is that we are able to show the potential of that approach; however operational application needs further developments in the cluster hardware including computational node power and inter-nodal

communication. The achieved scalability is satisfactory to 9-16 CPUs, when the balance between communication and processing time is still acceptable. Using more than 16 CPUs shows that communication overhead is much higher than gain from spreading of computations over higher number of CPUs.

The main advantage of running the COAMPS model in a grid environment is that at the same time different setups of the model might be run to feed different types of model in different spatial domains (waves and atmospheric pollution are setup in Baltic and Pontes power plant regions respectively).

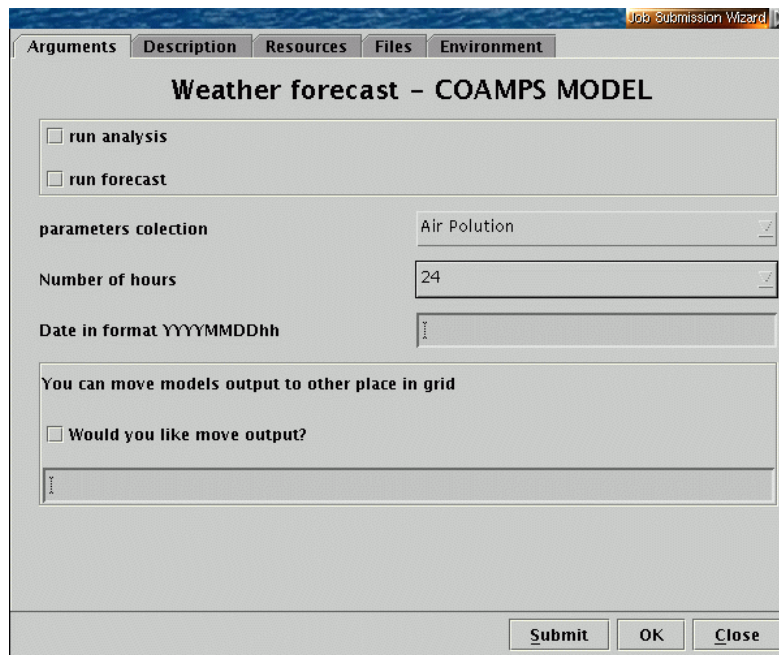


Figure 10: Atmospheric model GUI

The second application developed in task 1.4 is the air pollution model. The air quality application is based on STEM-II, an Eulerian air quality model used to simulate the environment of As Pontes Power Plant in A Coruna. This model is powered by meteorological data generated by atmospheric model and produces gaseous and aqueous concentration fields of each modelled species, reaction rates, amount of deposited species and ionic concentration. The model is run in parallel using MPICH-G2. Air pollution model needs forecasts of appropriate meteorological parameters over a specific area. In the designed application, the STEM-II model requests three-dimensional wind, rain and exchange coefficients data on 61 km by 61 km square grid with 1 km resolution horizontally, while vertically it requires 15 levels inside the atmospheric boundary layer and temporal spacing of one hour. This computational grid is centred at As Pontes power plant and is used by air pollution model to forecast the quality of the air.

With the aim of helping the experts to make decisions, in addition to generating fields with the exact concentration values of each point of the simulated mesh, we have developed a user-friendly interactive visualisation tool, which allows setting input/output parameters, and watch the concentrations of pollutants graphically (Fig. 11). This interface is interactive, meaning that technicians can change visualised parameters in real time during model run and follow the impact of these changes.

The last application ported to the grid environment in task 1.4 is the wave model application. The wave model application consists of Grid-enabled application, run from the portal interface. The wave model also depends on meteorological model fields.

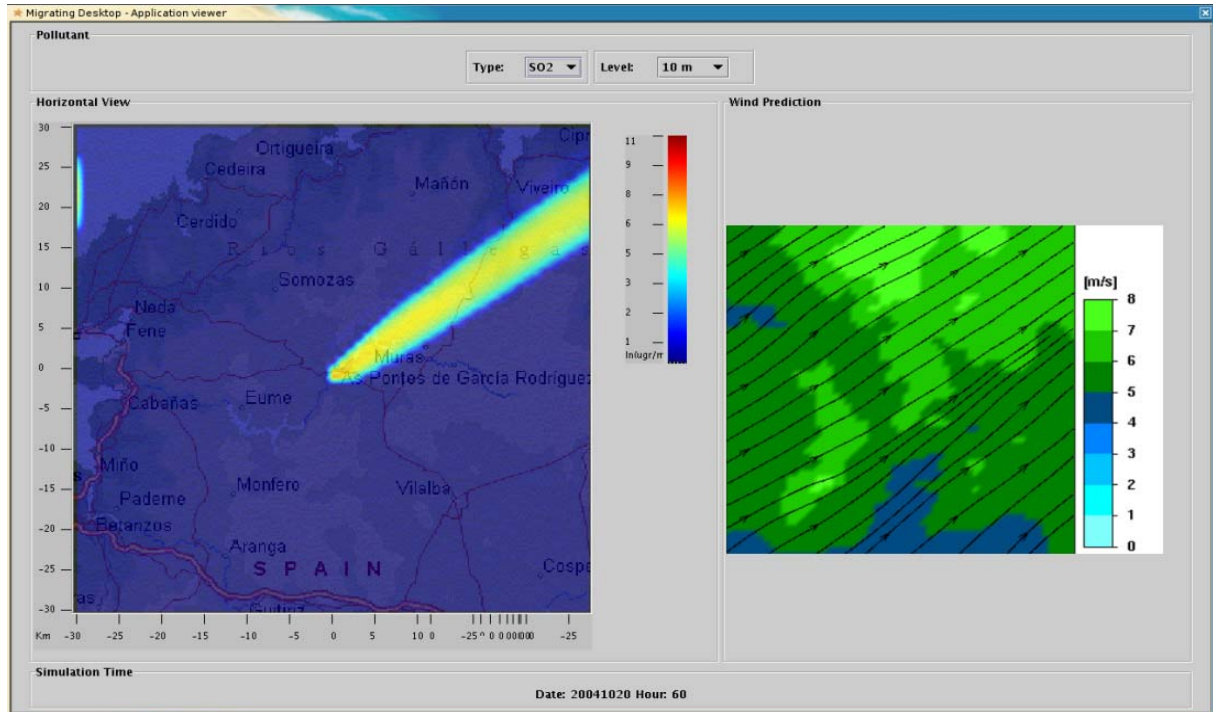


Figure 11: Example of an interactive output of atmospheric pollution model combined with atmospheric model data

Proposed presentation plan

There is plan to divide the show into two presentations. The first one should be a presentation of submission of atmospheric model jobs, both for Baltic and Pontes regions, in order to prepare driving forces for further demonstrations of air pollution and wave models. Because the atmospheric model runs take quite long time, we advise that presentations should be separated by several hours. If for organisational reasons this is not possible there will be forecast from the previous day ready to feed wave and air pollution models.

The first presentation will show the MD GUI of COAMPS model, with submission of different spatial domain tasks and monitoring of the job execution.

The second presentation will consists of interactive presentation of air pollution model with on-line output visualisation and portal wave model job submission and monitoring if required by reviewers.