A Geographical Information System for wild fire management

António Pina, António Esteves, Joel Puga, and Vítor Oliveira

Department of Informatics, University of Minho, Portugal
{pina,esteves,joel,VSPO}@di.uminho.pt

Abstract. The CROSS-Fire project focus on developing a grid-based framework for wild fire management using FireStation (FS) as a standalone application that simulates the fire spread over complex topography.

The overall software development is made of several components: client applications, which request geo-referenced data and fire spread simulation, Spatial Data Infrastructures (SDI), which provide geo-referenced data, and the GRID, which gives support to the computational and data storage requirements.

Herein we present the central Web Processing System (WPS) layer developed to support the interaction between all components of the architecture. This OGC-WS compatible layer provides the mechanism to access the Grid facilities for processing and data management and including all the algorithms, calculations, or models that operate on spatially referenced data, also mediating communication with the FS console.

We also describe the work that has been done to provide FS with dynamic fuel maps, by using an OGC-WCS suite of services and satellite data. This task complements the previous integration of dynamic data from meteorological stations using OGC-SWE services.

Key words: OGC-WS, Wild Fire Management, Grid Computing, Satellite Imagery

1 Introduction

The CROSS-Fire project[1] aims to develop a grid-based framework as a risk management decision support system for the civil protection authorities, using forest fires as the main case study and FireStation (FS) [2] as the standalone application that simulates the fire spread over complex topography.

The approach is based in an architecture that includes: information models, encodings, and metadata that represent the scientific knowledge associated to FS execution models and standards to enable the discovery and access of Web services, data repository, in-situ and satellite sensors, and data processing facilities.

The overall software development is made of several components: client applications, which request geo-referenced data and fire spread simulation, Spatial Data Infrastructures (SDI), which provide geo-referenced data, and the GRID, which gives support to the computational and data storage requirements. CROSS-Fire

uses the EGI/EGEE distributed computing infrastructure to provide raw technological capability provision, including data management and storage, access to meta-data databases and HPC.

On another hand, taking advantage of Open Geospatial Consortium (OGC) proposals for open standards for geospatial interchange formats, a standard-based SDI layer centered on Geoserver is being exploited to provide FS with static data, and to publish data for further processing, while provisions to dynamic data is make available by using an OGC-SWE compatible layer. The dynamic data includes meteorological information and satellite images, coming from sensors in weather stations (such as DAVIS Vantage Pro2) and sensors aboard satellites (such as Terra/Aqua MODIS).

The console of FireStation (CFS) is based on gvSIG, a full feature Open Source GIS desktop solution, funded by EC, which conforms to INSPIRE for managing geospatial information.

1.1 Open Geospatial Consortium

Open Geospatial Consortium (OGC) [3] provides several open standards to regulate geospatial content and services, GIS data processing and data sharing. Next, we describe its use in the context of CROSS-Fire.

- Web Map Service (WMS) provides georeferenced images (maps) with no other data associated to them, used as background maps for simulation.
- Web Coverage Service (WCS) is similar to WMS, but it allows the association of data to the image (one or multiple values per cell), used for example, to provide altitude maps, where the value of a cell represents the height at that point, as is the case of raster type data (terrain, fuel maps and DTMs).
- Web Feature Service (WFS) is used to provide representations of real life entities (roads, buildings, rivers, monuments, etc.) that have a wide array of data associated with them, such as: wind field and fire spread simulations results.
- Web Processing Service (WPS) is used to request execution of a process that works with geospatial data. It is the core layer of the CROSS-Fire platform.
- Catalog Service Web (CSW) provides a catalogue of meta-data especially designed to organize geospatial data, used to catalogue and organize geospatial data, and speed up the search for the sources of simulation data.
- Sensor Observation Service (SOS) provides access to geo-referenced sensors and their readings. We use it to get the readings of our meteorological stations.

1.2 CROSS-Fire Platform

The architecture (fig. 1) integrates the WPS core layer of CROSS-Fire and three infrastructures: i) a Distributed Computing infrastructure built on top of the EGI/Grid middleware, ii) a Spatial Data Infrastructure conforming to OGC-WS and SWE-SOS standards, and iii) a (third party) knowledge database organized according to Portuguese CP organization. In addition, a Human Interface layer provides all the functionalities needed for the interaction with the platform.
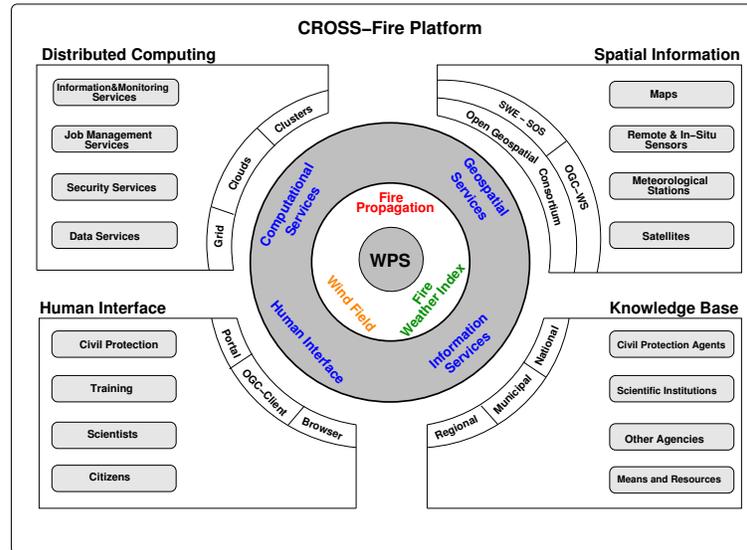
Fig. 1. CROSS-Fire Platform Architecture

The core of the platform is a 52North WPS layer [4] that deals with most of the functionalities of the dependent layers: Business Logic (BL), Computational Services and Geospatial Services, Information Service and Human Interface (fig. 2). The BL is an abstract layer containing all algorithms that provide the functionality of the application specific semantic, such as fire spread and wind field simulations.

1.3 FireStation

The FS is an integrated system aimed at the simulation of fire growth over a complex topography, that integrates a wind field generation the computation of the Fire Weather Index. FS requires three different types of input data to simulate fire propagation: (i) a description of the terrain features (e.g., morphology), (ii) meteorological data, namely the wind conditions affecting a selected region, and (iii) the initial conditions and some control parameters. The description of the topography and fuel characteristics of the terrain is registered in two files that store the altitude and the fuel type for each cell.

The wind field module receives as inputs a modified version of the terrain map called Digital Terrain Map (DTM) and the wind readings (direction and velocity) provided by the meteorological stations existing in the area of interest, returning as an output a map of the wind behaviour (direction and velocity) within the terrain limits. The fire-spread module receives as inputs the wind field simulation results, a fuel map and the respective fuel index (describing the characteristics of the different kinds of fuel), the terrain map and several inputs and parameters of the simulation (ignition points, fire barriers, stop conditions, etc...).

For each burned cell the simulation returns the values of: fire intensity in the line of fire, flame size, local fire velocity, flame life span, energy for unit of area and time of cell ignition.
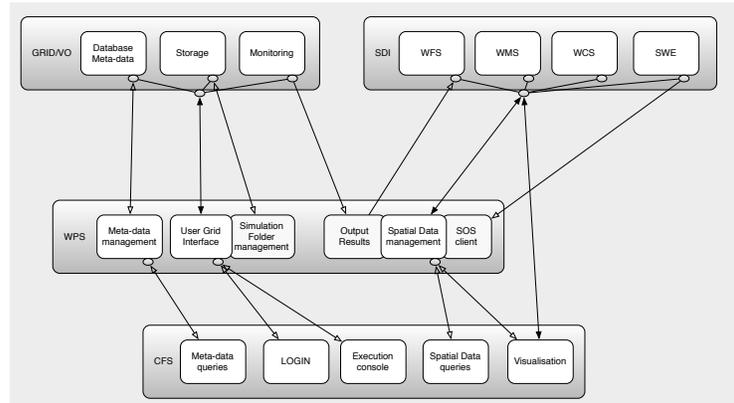

Fig. 2. WPS Layer

1.4 Meta-Data Management

In CROSS-Fire one expects not only to be able to run simulations but also to access and manipulate the results produced in the past. To accomplish these requirements, the platform must be able to register in a convenient way all input/output data used in the simulation. To achieve those goals, all the simulations input/output data, along with the meta-data that characterize the executions, such as the date, the execution time or the resources used, is recorded on a database server with GIS extensions. In order to have this data available on the Grid and, at the same time, to have access to the replication, redundancy and security facilities offered by the Grid, the database is interfaced via AMGA[5] data-base.

1.5 Spatial Data Infrastructure

The SDI of CROSS-Fire is composed by two mains components: a PostgreSQL database and the geographical data server Geoserver[6]. The database incorporates a Postgis [7] extension with several functionalities for geospatial data, compatible with Geoserver. All the input/output used/produced by FS is imported/made available in the database. As the original FS uses as input data files in a proprietary format, we needed to convert them to standard formats. The terrain and fuels maps are natively available in ARCgrid, an ASCII raster format widely used and compatible with most GIS software, so no conversion was needed. The modified DTMs are converted to this format; since they have only one value associated to

each cell. Other input/output data with more than one value associated to each elementary cell such as: wind field results, meteorological readings and fire spread results, are converted to SQL file format, to be stored in a PostgreSQL database.

Both types of data may be added to the geospatial server for post retrieve through the WFS and WCS standards. Whenever there is a need to get data from the Geoserver, several algorithms we developed are used to reverse the standard formats into the original format data used by the FS.

1.6 Related Work

Virtual Fire is a Web GIS platform to predict, simulate and visualize wildfires in real time [8].

EuroGEOSS is a large scale European integrated project for Environment Earth Observation that follows INSPIRE directive and builds an operating capacity for three areas: drought, forestry and biodiversity [9].

PREVIEW is an UN sponsored project to easily integrate and share spatial data on global risk from natural hazards [10]. It provides interactive and almost full access to data sets on 9 types of hazards at the global scale.

The EUMETSAT delivers weather and climate-related satellite data all days of the year. Instituto de Meteorologia, one of its members, operates the Land Surface Analysis Satellite Applications Facility (LSA SAF) whose objective is to increase benefit from EUMETSAT satellites.

GreenView, a Europe Grid based environment, offers tools for vegetation classification, satellite measurements calibration or temperature analysis and prediction [11].

ImazonGeo is a Web GIS based on a SDI, oriented to monitor Amazon forest, especially to avoid deforestation [12].

2 CROSS-Fire WorkFlow

The WPS layer comprises several modules or algorithms (the official term). It was developed in JAVA using the 52North framework that works in conjunction with Apache Tomcat. A CROSS-Fire specific XML data structure has been created to support most of the interactions between clients and algorithms, and the algorithms themselves. This structure, which is encapsulated on the standard XML used for the WPS request, comprises several fields used to register all the information associated with the inputs/outputs of FS, that are progressively filled by each of the different algorithms used for the launching and execution of the simulations.

In what follows, we present the CROSS-Fire platform workflow from the WPS point of view. The implemented algorithms are identified by names of gods and titans from Greek mythology.

2.1 Data Requests

Atlas - This algorithm is used to discover and return the geographical data provided through the WMS, WCS and WFS standards, that represent terrain and fuel maps.
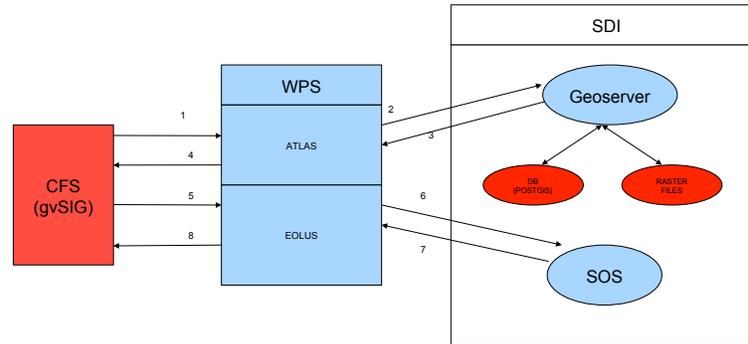
Fig. 3. Phase 1 - Data Requests

It takes as an input (1) a specific geographical area provided by the client through the XML data structure (see figure 3). It then (2) queries the Geoserver for all the maps within the specified input using a JAVA library that provides methods to search data, conforming to any of the OGC standards supported by the Geoserver (WMS, WCS and WFS). Each of these methods request the XML capabilities documents of each standard, that includes the bounding box of each layers, and then it looks for specific defined tags included in the map names to return (3) the correspondent maps types. In the case of WCS it is also necessary to search in the XML document returned by the DescribeCoverage request.

Finally, the algorithm returns through the XML data structure (4) the URL of each of the maps that are within the input area.

Eolus - This algorithm (see figure 3) return the meteorological readings provided by the SOS servers existent in a specified geographical area and in the time interval provided as inputs in a XML data structure (5). First, a method is used to discover (6) all the weather stations in the selected area. Then, a second method returns (7) for each founded weather station all the observations taken within the bounds specified by the time interval. If the first date specified in the interval is a future date another method will be used to obtain the latest read from each weather station.

For each station, the read values are inserted in the input data structure to be returned (8) to the client, by associating them with a generated identification (an hash value based on its name), its name, and its location.

2.2 Simulation Data Infrastructure

Ares - The algorithm configures the geospatial database, the Geoserver and the meta-data database to receive the results of the simulations (see figure 4). It then launches the jobs on the grid, through the following actions: (i) creation of the identifications of the different components required for the simulation, (ii) creation of a table for the simulation results, (iii) creation of a meta-data table for the

simulation meta-data, and (iv) copy the input parameters of the simulation to the grid and launching the correspondent jobs.
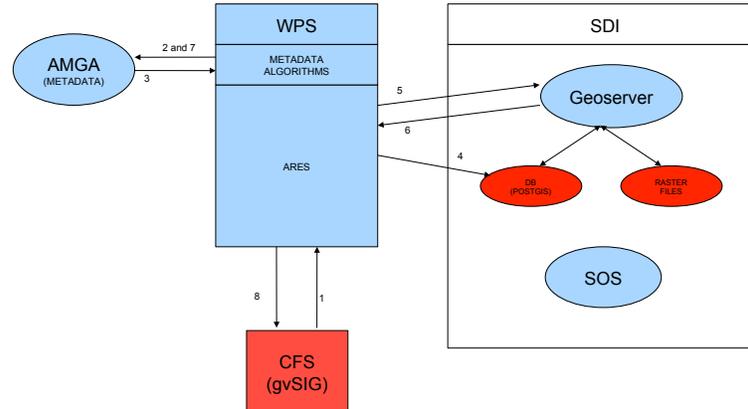

Fig. 4. Phase 2 - Simulation Data Infrastructure

As the previous algorithms, it receives (1) as input the XML data structure, which should contain all the data returned from the previous two algorithms, already customized by the client, to conform with each specific simulation requirements, and updated with other simulation parameters also provided by the client, such as stop conditions, ignition points, barriers, and so on.

Since this is a complex algorithm, in what follows the description of its functionality is divided in several phases:

Identifications generation: First, Ares verifies if the elements provided in the XML data structure are new, i.e. no previous simulation has used them, by checking if their values are equal to -1. If this is the case (2), the algorithm registers the new elements on the meta-data database by creating the elements in the database, and also defining their identifications, which are automatically generated by the Postgis database used to store the meta-data. Next, the new identificatiers are returned (3) and inserted in the XML data structure.

Space for the results: The identifications are then used to generate the tables (4) on the geospatial database that will contain the results of the simulations, through a JDBC connection. Each table identifies the type of the simulation and it's identification. Next, the tables are made available through WFS (5) using the Geoserver Representational State Transfer (REST) interface. Finally, the library returns (6) the location (URL) of the results that is inserted (7) in the meta-data database and the XML data structure is returned to the client (8).

2.3 Simulation Launching

This phase is divided in two steps implemented using bash scripts that: (i) copy the input data from the Geoserver to the grid and (ii) launch the jobs (see figure 5).

Ares - In the first phase, the algorithm obtains the data used by the scripts from the XML data structure, namely the simulation identificatiers and the URLs of the input data. Then, invoking a simulation folder script (SFM), it creates a folder structure in the Grid (1), which is identified by the identifiers, and copies the required data from the Geoserver into the Grid. It is at this point that the "reversers" are used to convert the data from a standard format to the format used by the FS executables.

Once terminated the copy of the input data, the job may then be submitted (3) using a simulation launcher (SL) script. All the information, related to the job returned by the Grid WMS, is registered in the XML data structure.

Hades - This algorism is used to gather the output from the simulations and insert it in the table created by Ares, and to update, at the same time, the related meta-data. Unlike the other algorithms, it does not receive as input the XML data structure, but a series of files, each one containing the simulation identifiers as header, followed by a series of lines, each one representing the simulation results for a specific cell in the map.
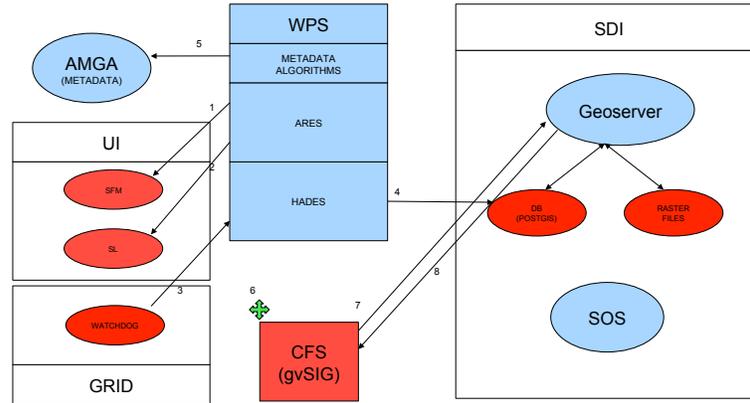

Fig. 5. Phase 3 - Simulation Launching

These files are provided by the Watchdog [13] script, which is launched in conjunction with the simulation, and runs in parallel to detect the availability of new results in the Grid, and to send them back to Hades (3). The algorithm may then insert the new data results on the corresponding table (4), previously created by Ares, using the simulation identifier. This is implemented by a Java library that inserts a batch of data in the geospatial database through a JDBC connection.

An internal meta-data algorithm, evoked by Hades, also updates (5) a simulation meta-data field in the AMGA database that states the time of the last

data received or the time of the simulation termination. This field may be used to determine if there is new data available to download (6) or the simulation is already terminated. The data results are available through the Geoserver in the URLs (7) previously provided by Ares. This data results may be used, for example, to visualize the progression of simulations.

3 Integration of Satellite Dynamic Data on FireStation

Given the potential of data that may be acquired from the satellite imagery we decided, to complement the integration of dynamic information from weather stations with the dynamic data/imagery obtained from Terra and Aqua satellites, in particular, data from MODIS [14] and ASTER [15] sensors. Since the MODIS' set of products is broader and more freely available than the ASTER's one, we opted for MODIS data. The objective is to automatically improve and update the available static fuel maps using the following satellite data: (i) updated vegetation, (ii) recently burned areas, and (iii) land cover type. The implemention is based on the access to geo-referenced images using the OGC-WCS standard [16] and its Web Coverage Processing Service extension (WCPS) [17][18].

The WCS standard defines basic extraction operations, such as spatial, temporal and band subsetting, scaling, reprojection, and format encoding. Although WCS does not offer a powerful processing capability, it is declarative (describes the results rather than the algorithms), it has a safe evaluation (impossibility of a request to keep the server busy infinitely), it is optimizable (capability of the server to rearrange the request to produce the same result faster), and its explicit semantics allows machine-machine communication and reasoning. The OGC WCS-Transactional (WCS-T) is a WCS 2.0 extension that allows to insert and update coverages stored on a WCS-type coverage server. WCS-T specifies an additional `Transaction` operation.

WCPS is a WCS 2.0 extension that provides an additional `ProcessCoverages` operation. WCPS defines a flexible interface for the navigation, extraction, and ad-hoc analysis of large, multi-dimensional raster coverages, that allows to combine several coverages in one evaluation. Each coverage is optionally checked first for fulfilling some predicate, and it contributes to the result list only if the predicate evaluates to true.

3.1 MODIS Data for Dynamic Fuel Maps

In what follows we describe three MODIS land products used to compute updated and improved fuel maps.

Burned Areas - Produced from both Terra and Aqua MODIS daily surface reflectance, MCD45A1 is a monthly L3 gridded 500 m product that approximates the date of burning, and maps the spatial extent of recent fires. The most relevant MCD45A1 layer, among the 8 available ones, is the `burn date`. This field approximates the Julian day of burning (1-366) from eight days before the beginning of the month to eight days after the end of the month, or a code indicating especial cases.

Land Cover Type - These MODIS products containing multiple classification schemes describe land cover properties derived from one year of Terra and Aqua observations. The MODIS Terra and Aqua land cover type yearly L3 global 500 m sinusoidal grid product (MCD12Q1) incorporates five different land cover classification schemes. As an example, the IGBP scheme identifies 17 land cover classes containing: 11 natural vegetation classes, 3 developed and 3 mosaicked classes.

Vegetation Indices - Using MODIS blue, red, and near-infrared reflectances, and masking water, clouds, heavy aerosols, and cloud shadows, vegetation indices are computed daily. Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI) are among these indices. For example, the global MOD13A1 product is provided every 16 days at 500 m spatial resolution as a gridded L3 product in the sinusoidal projection. Vegetation indices are used for global monitoring of vegetation conditions and are used in products displaying land cover and land cover changes.

3.2 Dynamic Fuel Map Computation

Next we summarize the steps needed to compute a fuel map with the data presented in 3.1.

First, the data is downloaded from Nasa/MODIS database, as tile-based Hierarchical Data Format (HDF) files (figure 6), using a specifically developed FTP downloader application. HDF is a data format designed to facilitate sharing of scientific data in a heterogeneous computing environment [19].
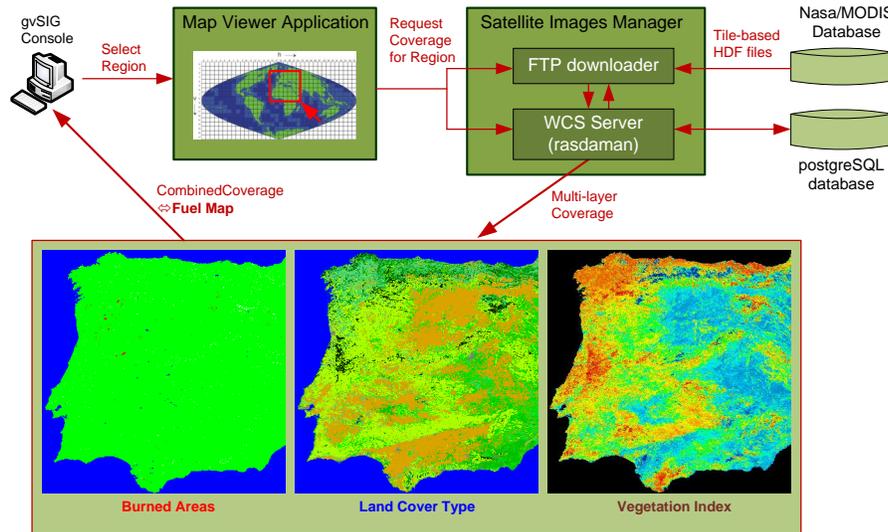

Fig. 6. Illustration of the computation of dynamic fuel map with data from MODIS.

Then, the layers of interest are extracted from each file and merged to form a single raster image for a multiple tile region. This image and its metadata are inserted in the WCS server database via the WCS-T `Transaction` operation.

The implementation is based on the WCS 2.0 available in the `rasdaman` project [20] that also includes a database management system (DBMS) layer that supports multidimensional arrays of any size and dimension and over freely definable cell types. At the bottom of the rasdaman architecture, arrays are decomposed into smaller units which are maintained in a conventional relational DBMS (postgreSQL).

To obtain a fuel map with the data stored on WCS, we may use both WCS and WCPS requests: (i) to apply filters (subsetting) to the coverages for getting the region of interest, (ii) to convert the resolution, for equalizing the resolution of the various types of information needed to calculate the fuel map or generating the fuel map with the resolution required by FireStation, (iii) to combine the land cover type coverage layer, with the vegetation index layer and the burned area layer, and to produce a new fuel map, and finally (iv) it is necessary to return the fuel map in ArcInfo ASCII grid format, to be imported into FireStation.

4   Work in Progress

Authentication - Only some preliminary research was been done on the available authentication and credential delegation software to access both the SDI and the GRID using X509 certificates. Gridsite [21] is one of the candidates to implement this functionality, because it allows the delegation of Grid credentials, and also the use of these certificates to control the access to applications served by Apache Tomcat (Geoserver, WPS, etc.)

Catalog System - We have a working installation of CSW 2.0.2 standard implementation using Degree CSW [22], natively compatible with Apache Tomcat, that we plan to integrate in the WPS's Atlas algorithm.

Portal - A simple Web portal was developed using JavaScript, HTML, and Openlayers [23], to provide an OGC conformant Web GIS client. The portal allows the user to visualize the results of previous simulations on top of a background terrain/street map.

Dynamic Fuel Maps - Rasdaman is an evolving project for which we are contributing by adding to WCS suite the support for multi-layer coverages and the associated file formats.

Acknowledgements

References

1. CROSS-Fire, *Collaborative Resources Online to Support Simulations on Forest Fires*, FCT GRID/GRI/81795/2006
2. Lopes, A., Cruz, M., and Viegas, D. *FireStation - an integrated software system for the numerical simulation of fire spread on complex topography*,Environmental Modelling and Software, 17(3):269285, 2002
3. Open Geospatial Consortium, `http://www.opengeospatial.org/`
4. Simonis I., Wytzisk A., Echterhoff J. *Sensor Web Enablement: The 52North Suite*, Proc. of the Free And OSF for Geoinformatics , Lausanne pp. 11-15, (2006).
5. Santos N., Koblitz B. *Metadata services on the Grid*,Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research, 2005
6. GeoServer, `http://geoserver.org/display/GEOS/Welcome`
7. Postgis, `http://postgis.refractions.net`
8. Kalabokidis K., Kallos G. *Web GIS platform for forest fire management*, Virtual Fire Final Review and PR Event, Mytilene, Greece, 2010.
9. Vaccari L., Nativi S., Santoro M. *Deliverable D.2.1.1. Report on requirements for multidisciplinary interoperability*, Project EuroGEOSS, a European approach to GEOSS, 2010.
10. Giuliani G., Peduzz P. *The PREVIEW Global Risk Data Platform: a geoportal to serve and share global data on risk to natural hazards*. Natural Hazards and Earth System Sciences, Volume 11, pp. 53-66, 2011.
11. Mihon D., Bacu V., Gorgan D., Mszros D., Gelyb G. *Practical considerations on the GreenView application development and execution over SEE-GRID*. Earth Science Informatics, Volume 3, Number 4, pp. 247-258, Elsevier, 2010.
12. Souza C., Pereira K., Lins V., Haiashy S., Souza D. *Web-oriented GIS system for monitoring, conservation and law enforcement of the Brazilian Amazon*. Earth Science Information, Volume 2, pp. 205-215, Springer, 2009.
13. Bruno R. *A watchdog utility to keep track of job execution*, INFN, `https://grid.ct.infn.it/twiki/bin/view/PI2S2/WatchdogUtility`
14. United States Geological Survey (USGS). *MODIS Overview*, October 2008. `https://lpdaac.usgs.gov/lpdaac/products/modis_overview`.
15. Abrams M., Hook S., Ramachandran B. *ASTER User Handbook, Version 2, Jet Propulsion Laboratory, California Institute of Technology*, August 2002. `http://asterweb.jpl.nasa.gov/documents.asp`.
16. Whiteside A., Evans J. *Web Coverage Service (WCS) Implementation Standard*, Open Geospatial Consortium Document 07-067r5, March 2008.
17. Baumann P. *Web Coverage Service (WCS) - ProcessCoverages Extension*, Open Geospatial Consortium Document 08-059r3, March 2009.
18. Baumann P. *The OGC Web Coverage Processing Service (WCPS) Standard. Geoinformatica*, July 2009.
19. Nasa Goddard Space Flight Center (GSFC). *Science Data Production (SDP) System Toolkit*, April 2010. `http://newsroom.gsfc.nasa.gov/sdptoolkit/toolkit.html`.
20. Baumann P. (rasdaman GmbH). *The Rasdaman Project Documentation*, 2010. `http://kahlua.eecs.jacobs-university.de/trac/rasdaman/wiki/Documentation`.
21. GridSite, `http://www.gridsite.org`
22. Deegree CSW, `http://download.deegree.org/deegree2.4/docs/csw/html/deegree_csw_2.0.2_documentation_en.html`
23. openLayers, `http://openlayers.org`