

A SERVICE ORIENTED APPROACH TO MODEL A GRID SYSTEM FOR THE CIVIL PROTECTION

Abeti L. ^{†,§}, Ciancarini P. [§], Moretti R. [§]

[§] *Università di Bologna*¹ [†] *Regione Marche*²

Keywords: Grid, MDA, SOSE, Semantic Grid, Civil Protection, Emergency Response Grid

Abstract

This paper shows how some technologies based on the Model Driven Architecture, Service Oriented Architectures and Grid systems can be exploited to design and develop a Grid infrastructure useful for Civil Protection systems. Current MDA technologies can be employed in Grid Systems to overcome some interoperability weaknesses and to obtain a uniform interaction method by means of the "service" concept. A new approach named Service Oriented Software Engineering (SOSE) is presented together with a supporting tool. Such a tool named MOdeling TOol for Grid and Agent Services (MOTO-GAS) allows to model services and their interactions by means of some metamodels. We apply the MOTO-GAS technology to model the Grid-Care system. Grid-Care is aimed to develop an "infrastructure of services" needed to compose Service Oriented Applications for Civil Protection systems. In particular we represent how to model two complex scenarios to support meteorological forecasting and storm tracking. This work opens new perspectives for computational Grid employment in the context of e-government and business continuity.

Introduction

Convergences between the Grid, the Model Driven Architecture (MDA) and the Semantic Web are leading promising research field in computer science today. In particular a model driven approach to the Semantic Grid can supports agents in a future e-science infrastructure.

The Grid and Web Service convergence realized by the WSRF (Web Service Resource Framework) makes possible to apply object-oriented design practices to Service Oriented Architectures (SOAs). It identifies in the "service" concept a basis for current technologies

¹Facoltà di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Scienze dell'Informazione Mura Anteo Zamboni, 7 - 40127 Bologna - ITALY
GPS: N44.29.826 E11.21.375 (WGS-84) Tel: +39 051 209.45.16 Fax: +39 051 209.45.10
{abeti,ciancarini,moretti}@cs.unibo.it

²Sistema regionale Protezione Civile e Sicurezza Locale – Centro Funzionale regionale per la meteorologia, l'idrologia e la sismologia Strada Cameranense, 1 – 60029 Ancona – ITALY
GPS: N 13.52461 E 43.57421 (WGS-84) Tel: +39 071 806.7723 Fax: +39 071 806 7709

interactions. It is expected that new software engineering practices will be used to the design of service-based applications.

Service-orienting is a common vision for many fields of Computer Science. SOAs can be envisioned as enhanced middlewares not assuming object orientation where components are meant as loosely coupled services. This approach, enriched by design supporting tools, allows a new Software Engineering prospective bringing to the vision of a Service Oriented Software Engineering (SOSE).

It is likely that the real application of the Service Oriented Software Engineering will be in Semantic Grid (the application of Semantic Web intuitions to the Grid world). Both agents and Grid systems consist of dynamic and stateful services. SOSE is a mean to integrate these two visions inside a single modeling framework based on the higher-level abstraction of service. It has been stated that “every agent can be considered a service” [1] thus in SOSE we consider agents no more than special active kind of Grid service resources.

A Model Driven Semantic Grid can be considered the architecture needed to support agents in an e-science infrastructure. A tool named MOdeling TOol for Grid and Agent Services (MOTO-GAS) [2, 3] has been developed. This tool using the Model Driven Architecture (MDA) approach allows to model services and agents by means of some metamodels. MOTO-GAS can be used to represent and manage in a unique framework the three main components of a Model Driven Semantic Grid: Web Services, Stateful Web Services (WS-Resources) and Agents.

Following the SOSE intuition, MOTO-GAS uses Model Driven Architecture (MDA) to model services and their interactions (by means of meta-models for WSDL, WSRF, Business Process Execution Language for Web Services (BPEL4WS) and Classifier Advertisement (ClassAd)).

We apply SOSE in a real project proposed to develop a Grid for the Italian Civil Protection (named Grid-Care). The MOTO-GAS tool is employed to show how complex application can be easily modeled in a Grid system to manage hydro-geological risk and to assist weather forecasting.

Grid-Care defines an architecture for a “national Grid” aimed to support issues related to meteorology and hydro-geological risks in the Civil Protection system. We show how Grid-Care allows to reach strict correlation between centers distributed all over the national territory and highly heterogeneous (both in their business organizations and in used technologies).

Grid-Care is aimed to produce a Grid support infrastructure for all task of Civil Protection Functional Centers. Its principal needs are related to the sharing of the following resources: Computational power; Storage resources; Databases resources; Data from meteo-hydrological sensors; Maps and Graphs.

We show how to apply MOTO-GAS for the design of two complex applications using these services, and represent them as a service as well. These applications are: A Meteo Forecasting Support System and a Storm Tracking System.

Following the SOSE approach an application service model has been depicted for Grid-Care. In this model we represent the MeteoForecastSupport and the StormTraking service. MeteoForecastSupport permits to use outputs of the other Grid-Care services simultaneously to support weather forecasting activities. StormTraking is an Artificial Intelligence(AI) application which permits to put meteorological observation and prediction together with the hydro-geological response of the territory and notify if risks situations occur.

Once we create this first Grid VO to interconnects Civil Protection Centers; other applications can be carried out straightaway using a small set of core services. Epidemiological simulation for bacteriological attacks; earthquake modeling, climatological studies and other usages of the Grid can be explored to improve virtual collaboration and risk management in Civil Protection Systems.

The following section describes the Service Oriented approach used to define the Grid-Care architecture together with the MOTO-GAS tool that allows service modelation. Next two sections describe the Grid-Care project and depict an application service model for the Grid-Care infrastructure. Finally we furnish a complete exposure of three fundamental services developed in Grid-Care: Meteorological Sensor, Storm Tracking and Meteorological Forecast Support.

New outlooks for software development: Service Oriented Applications

Interoperability between different middleware platforms (in space and in time) is not completely realized today. MDA attempts to address this need starting from the viewpoint of models and model-specifications. This seems to be a good prospective from which the problem can be solved but is not enough. Another more specific solution is given by Service Oriented Architecture (SOA) which is "an approach to distributed computing that regards software resources as services available on a network" [4].

The real improvement of SOAs is due to the Web Service orientation of such systems. Many middleware weaknesses disappear with Web Services. Web Services allow Service-Oriented Architectures standards-based and with loosely coupled services. Unfortunately distributed applications cannot be developed only through Web Services because a Web Service always needs a concrete software agent to work. Though Web Service cannot be considered as a middleware, another emerging middleware shifts the emphasis on resource sharing rather than on simple remote object coordination and also uses loosely coupled components: the Grid.

The convergence between the Grid, Web Service and Middleware, envisioned as a Service Oriented Architecture, supported by the versatility of Model Driven Architecture is a new vision of how to apply distributed system to applications by using the service abstraction. The concept of service is proposed as a new abstraction for software development. In the same way, the Service Oriented Software Engineering (SOSE) is proposed as a new hypothetical approach to the definition and development of enterprise service oriented applications.

In this context the user can define a set of stateless "Services" and stateful "Resources" which can be composed together and employed over the internet, much like object are composed and used in Object Oriented paradigm. This approach joints two not-properly-middleware technologies, Grid and Web Services, and it results in a new more powerful and flexible middleware.

The role of Model Driven Architecture is crucial in this framework. It permits to integrate several technologies and maintains complex applications within an easy architecture. It further provides the high-level approach needed in services development. Another advantage in using MDA is that it allows the adoption of a widely recognized standard for application modelling: the Unified Modelling Language (UML).

The MOTO-GAS tool can be a valid prototype to support Service Oriented Software Engineering. It allows to model both stateless and stateful services (hence Grid Services) in a Platform Independent Model (PIM) of a system. In other words it can be described a "service model" for the application that can be instantiated like a UML Class model.

MOTO-GAS applies the Model Driven Architecture approach to the modeling of some aspects of a Grid system and is able to produce part of the XML documents and Java interfaces that must be used in the implementation of the represented system. The tool permits to model WSDL and WSRF specifications in order to allow the concrete definition of Grid Services, Web Services and Agents together with their interactions.

Figure 1 shows how a simple resource is represented in MOTO-GAS. It defines a set of attributes needed for interaction and state definition together with a set of methods which defines the portType of the Service.

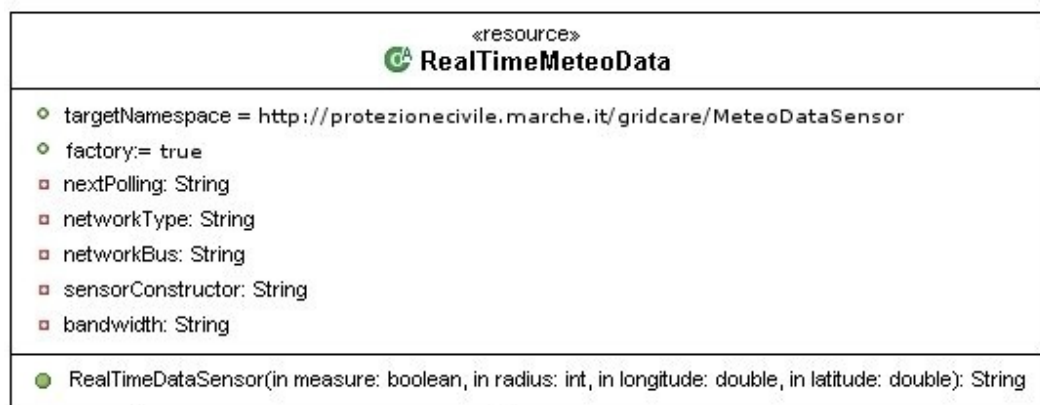


Figure 1: Source model for a single resource modeled with UML

Grid Care: A Grid for Civil Protection

Effects of historical earthquake, tsunami and hurricane would have been significantly reduced if an efficient world-wide Civil Protection alert system existed. Since a Civil Protection world-wide system is today a great challenge, governments are working hard in Italy and in Europe to better enable citizen defence for national and international natural calamity.

A project has been initiated in Italy for issues related to meteorology and hydro-geology³. The project is aimed to create a number of Functional Centers (CF) that continuously (24 hours a day) elaborate meteorological and hydro-geological forecast for an assigned region. This project will be a leading experience for others EU countries.

A strict correlation between civil protection centers, distributed all over the national territory imply the need for great levels of resource sharing.

³ Italian Parliament's Acts 267/98 and 365/2000

Centers are highly heterogeneous, they differ either because of the technologies used or because of the administrative organization. There are CF affiliates to University or research centers (e.g. CETEMPS - Abruzzo, CIMA - Liguria), others included in Minister departments or Regional agencies (e.g. National Civil Protection Department - Italy, SPC Regione Marche - Marche) and others organized in semi-private forms (e.g. ARPA Agency - Emilia Romagna). Moreover all CFs must actively interact with other organizations such as big companies, military structures and mass media agencies. Despite "structural" lacks of homogeneities between participants, the functional centers system is a strategic organization and must be created starting from these actors.

Laying aside business organization issues also technical integration is not a simple task in these centers. Usually, they do not have the complete control of their computer networks (they are part of grater organization), furthermore, they use different operating systems (Unix, Linux, MacOS, MS Windows) and different database management systems (Oracle, Microsoft SQL Server, proprietary Databases, semistructured Databases and data warehouses). Discrepancies occur also in instruments. Meteorological and hydrological sensors for instance are built by different manufacturer, take measures at different time lags and transmit data using different network technologies (GPRS, UHF Radio, SHF Radio, PSTN).

Stated such scenarios, it will be a great idea to build a Grid for the Functional Centers project. Without the attempt to change this great level of heterogeneity, a computational Grid could straightforward solves technical issues and connects all Centers in a collaborative Virtual Organization (VO).

A Grid for Italian Functional Centers envisioned as Virtual Organization (VOs) may solve many share and interaction problems and allow a more efficient usage of resources. For instance there are tens of gigabytes data received from the Meteosat satellites (10Gb/day) which are currently stored in every CF database and rarely accessed. A Grid could logically centralize the management of data and optimize resources for every center through sharing of databases and storage. Another example is the need of computational power to elaborate models. Meteorological and hydro-geological models could run on a Civil Protection Grid and give results faster and better than now.

Once we create this first Grid VO that interconnects Civil Protection Centers, other applications can be carried out straightaway: epidemiological simulation for bacteriological attacks; earthquake modelings, climatological studies and other usages of computational Grid can be explored to improve virtual collaboration.

The Functional Center in the Marche Civil Protection is planning a project called Grid-Care with the intent to obtain, in the future, a Civil Protection Grid scenario like the one depicted here.

Application Service Model for Grid-Care

Grid-Care is a project aimed to produce a Grid support infrastructure for Civil Protection Centers in Italy. Grid-Care improve sharing of the following resources:

- Computational power;
- Storage and Database resources;
- Data from meteo-hydrological sensors;
- Maps and Graphs

Once services for these basic requirements are defined, more complex applications can be developed to support risk management and monitoring basing on them. In this paper it is showed how to apply MOTO-GAS for the design of two complex applications using these services, and represent them as a service as well: A Meteorological Forecasting Support System and a Storm Tracking System.

For the Grid-Care project ten fundamental services are identified, only in part to be realized from scratch. These services can run (or live) in the Grid infrastructure and be used independently from some applications or through web user interfaces. The service model of Grid-Care is made up of:

- *RealTimeMeteoData*: This is a service implemented as a WSRF Resource. It allows to access remote meteo-hydrological sensors to obtain current measures. It has a single portType named RealTimeDataSensor which allows to obtain data from specific sensors in a defined circular area.
- *LokalForecast*: This Web Service is a referenced service which furnishes output data of the German LokalModell developed by the COSMO project (the terms “lokal” and “modell” are German words). This model requires a lot of computational power to run. It takes in input large scale meteorological forecasting from a General Model (named GME) and computes rainfall prediction on a smaller scale.
- *GridStorage*: This is a WS-Resource that implements a storage manager inside Grid-Care. It is referenced and may be usually developed directly from the platform specific framework (for example in the Globus Toolkit 4 we can use OGSA-DAI services).
- *GridComputationPower*: As for the GridStorage service above this is a general service used to obtain computational power. Like for storage it is referenced and we can think it may be developed using some framework existent services such as Globus Toolkit 4 WS GRAM.
- *SatelliteObservation*: It is another referenced service that accesses a service that returns Meteosat8 satellite data and maps.
- *LokalRiskDatabase*: It is a stateless implemented Web Service which returns historical data on landslides, inundations, seaquakes, tornados, meteorological alerts and hurricanes. The database used by the LokalRiskDatabase service is continuously updated by the Risk Event Information System (REIS) of the Marche Region which allows capillary knowledge acquisition from local authorities.
- *GeneralModelForecast*: It is a referencedWeb Service that returns general forecasting model output. General Models do the same work of local models but for a wider area and obviously with less precision. It furnishes long term weather forecast data that can be used as the input of small scale weather or hydrological models.
- *FloodModel*: This referenced resource returns the output of a model that computes the hydrograph produced by a given precipitation event in a selected river cross-section. It uses the Mike-DRiFt model, a semi - distributed event - based rainfall runoff model used to describe single runoff event.
- *MeteoForecastSupport*: Using services described above, a meteorological forecast support system can be realized as a WS-Resource. This is a complex service that permits to use all services defined in Grid-Care simultaneously and in a unique architecture.
- *StormTracking*: It permits to relate meteorological predictions and observations with hydro-geological models and notify if risks scenarios occur. Meteo ForecastingSupport and StormTracking are better discussed in following paragraphs.

A complex application like this can be represented, modeled and in part realized and maintained using MOTO-GAS and Service Oriented Software Engineering (SOSE). Figure 2 represents this

application scenario in SimpleUML (a simplification of the Universal Modeling Language) it uses three stereotypes: <<service>> for a fresh definition of a Web Service; <<resource>> for a new stateful Web Service (WS-Resource); <<refservice>> for referred services and <<refresource>> for referred resources.

In the following sections we give details of the RealTimeMeteoData, StormTracking and MeteoForecastSupport WS-Resources.

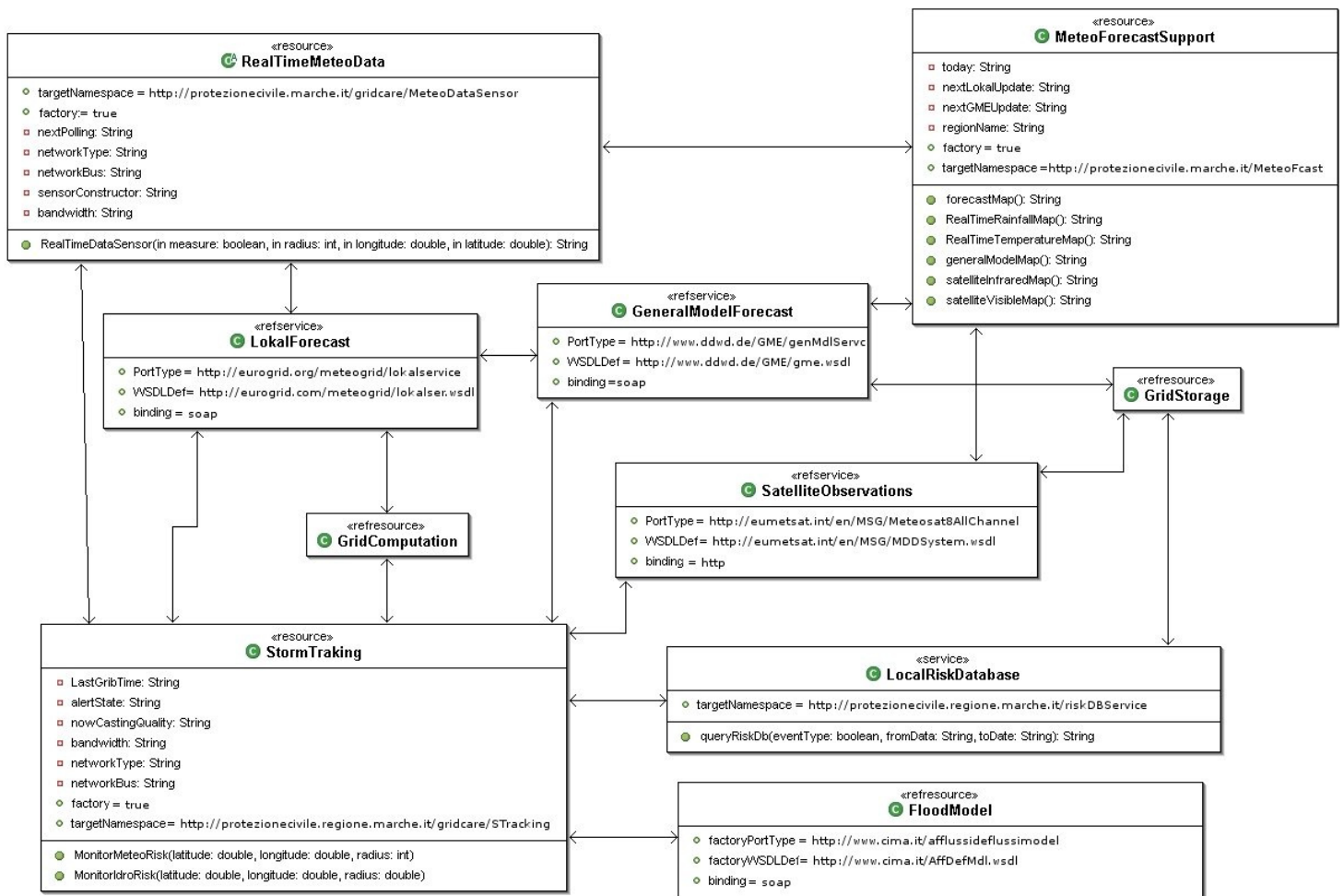


Figure 2: The Grid Care Service Model

The Meteorological Station WS-Resource

In this section we give a complete exposure of the *RealTimeMeteoData* WS-Resource modelled in figure 2.

Real-time data is very useful in meteorological nowcasting forecast systems, where predictions for the next 3-6-12 hours are given to prevent meteorological and hydrogeological disasters. In the service scenarios of figure 2 we can see how other services and resources are connected to this Web Service to obtain such information.

The *MeteoForecastSupport* resource needs to access this service to know the instantaneous air temperature in a land or the level of a river or again the quantity of rainfall during a storm. It then represents this data on graphs that a meteorologist can interpret and use.

Also the *StormTracking* resource, in its artificial intelligence elaboration, can ask for current observations in a specific area. Requests can also be made by the *LokalForecast* service to refine its model or to set initial conditions for a new local elaboration.

In a Civil Protection Grid, instrument share has great importance. We choose to describe such scenarios because it is meaningful for understand power of knowledge sharing in a e-science infrastructure. It also provides an example of a virtual instrument sharing according to the GRIDCC approach of remote sensing: "virtual instrument as special kinds of resources".

RealTimeMeteoData uses two types of state information: "metadata on the resource" or "data for which the service is responsible" (including state data). For instance *SensorConstructor*, *networkType* and *technology* are metadata about the meteorological sensor resource; while *bandwidth* and *networkBus* are state data which dynamically evolve during the execution of message exchanges. We choose to manage other information such as sensor type (measure) and observation area (radius, longitude and latitude) using the service port type. We make this choice only because we retain it useful in the context of these application, obviously sensors could be modeled in other ways.

RealTimeMeteoData takes great advantages from *WebService* standard. This simple service solves one of the greatest issues for CFs interoperability that is exchange of real-time data coming from different manufactured instruments, measured with different technologies, received at different time leaps and stored in different database management systems.

The MeteoForecastingSupport and StormTracking WS-Resource

The *Grid-Care* service application infrastructure shows how useful applications can be easily modeled using the *MOTO-GAS* and *SOSE* principles. The *Meteo-ForecastingSupport* service and the *StormTracking* are two fundamental applications for the *Functional Center* project.

Application like these will be developed with much lower costs and less effort once the other eight services are implemented.

MeteoForecastingSupport service is an application which returns a number of meteorological maps. Maps are rendered both from observed data and prediction model data:

- *forecastMap()* returns two maps for local rainfall and temperature forecasting in the region stored in the state (it uses the *LokalForecast* service)

- *realTimeRainfallMap()* returns the last observed rainfall for the region in a picture (it uses RealTimeMeteoData)
- *realTimeTemperatureMap()* returns the last observed temperature in the region.
- *generalModelMap()* returns the rainfall and temperature forecasting in the entire country as modeled by GeneralModelForecast service.
- *satelliteInfraredMap()* and *satelliteVisibleMap()* return respectively satellite images rendered with infrared sensors and satellite images from visible sensor data.

Attributes in MeteoForecastingSupport service are all state data and maintain: the current date (*today*); the time when the next output from the General and the Local Models (*nextLocalUpdate*, *nextGMEUpdate*) is expected and the predefined name of the region in which meteorologist are interested (*regionName*).

In the context of CF Project an application named STORM has been developed for the purpose of assist Meteo Forecasting. It will be very interesting to start from this software to give out a Client tier for the MeteoForecastingSupport service.

The *StormTracking* application is more complex from an implementation point of view. It tries to model a very challenging problem. Rainfall forecasting is not solved at small basins hydro-geological process scales; so it is very hard to relate meteorological situations to hydro-geological risks (storm tracking). Scientist normally use probabilistic and mathematics models to solve this problem. With this service we propose to model a storm tracking system using Artificial Intelligence algorithms. StormTraking relates meteorological observation and historical information for the sake of predict hydro-geological risks.

The service also monitors meteorological risks (tornado, hailstorms, storms etc.) for a given area. The state of this resource maintains some information needed in the forecasting that may also be used by other services for real time computations. It makes a wide usage of lokalRiskDatabase service to extract historical information. Also the FloodModel WS-Resource is widely used by the StormTracking service; it furnishes the output of a rainfall runoff model named Mike-DRiFt which simulate the discharge in a river cross section.

All information acquired from the other services are then processed with neural nets and adaptive algorithms to predict effects of the storm in the small basin scale.

Since it is expected that StormTracking will have a complex implementation; resources needed for this very complex service are all easily accessible in the modeled Grid. Moreover, the availability of a computational power service and a centralized storage service permits to realize this application in a more cost effective way.

Conclusions

Differently from other technological trends, Computational Grids do not need only technical solutions to be realized. Economical and political factors are equally important. This is due to the fact that the Grid is an infrastructure that is very similar to the Internet and its development will involve and interest everybody, just like the Internet. Fortunately, governments take this argument into serious consideration and invest millions of euro a year in this specific research field. Governments can take significant advantages from Grids which can be used not only for economic Return Of Investments but also to save citizen lives as we have shown with Grid-Care. In this context we might refer to Grid-Care as a valid example of the employment of computational Grid in e-Government.

The service abstraction and service-oriented view of software is an emerging approach which meets consensus in various fields of research. In this article we adopt a perspective that we call Service Oriented Software Engineering (SOSE). This perspective seems to be a good way to integrate new design techniques with recent middleware technologies. The vision is appreciable since it is realized in a framework based on the Grid-Service infrastructure, which is a kind of SOA.

The modeling tool used in this paper (MOTO-GAS) gives some substance to our perspective of SOSE. It allows to model both the Web Services and the Grid Services. It also permits to define business processes for those services where behavioural aspects are also meaningful (thus in future for agents).

Finally, the Grid-Care model is represented in a very straightforward way through a working Platform Independent Model (PIM) represented in UML. The support of this MDA architecture allows to maintain the Grid-Care infrastructure in a flexible and effective way .

References

- [1]FJK04] I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why Grid and agents need each other. In Proc. 3rd Int. Conf. 13 on Autonomous Agents and Multi-Agent Systems, New York, USA, 2004.
- [2]Abe05] L. Abeti. Service Oriented Software Engineering in Grid systems. Master's thesis, University of Bologna, 2005.
- [3] ACMM04] Andreozzi S, Ciancarini P, Montesi D, and Moretti R. Towards a Meta-modeling Based Method for Representing and Selecting Grid Services. In Proceedings of the International Conference on Grid Services Engineering and Management (GSEM'04), pages 78{93, Erfurt, Germany, Sep 2004.
- [4]lo03] Jason Bloomberg. Principles of soa. Application Development Trends Magazine, March 2003.

Author Biography

LUCA ABETI⁴ has a master degree in Computer Science and he is IT Official in the Marche Region Civil Protection Office (Italy)

PAOLO CIANCARINI is full professor in Computer Science in the University of Bologna (Italy)

ROCCO MORETTI is a PhD researcher in the area of Software Engineering affiliated to the Department of Computer Science of the University of Bologna (Italy)

⁴ Corresponding Author:

Luca Abeti
Regione Marche Protezione Civile - 1, via Gent.le da Fabriano 60100 Ancona
Tel# +39 0736 492565 Mob# +39 335 7437431 Fax# +39 071 806 7709
EMail Addresses abeti@cs.unibo.it - luca.abeti@regione.marche.it