MapCenter: An Open Grid Status Visualization Tool

Franck Bonnassieux
UREC CNRS
ENS Lyon, France
franck.bonnassieux@ens-lyon.fr

Robert Harakaly
UREC CNRS
ENS Lyon, France
robert.harakaly@ens-lyon.fr

Pascale Primet
RESO INRIA
ENS Lyon, France
pascale.primet@ens-lyon.fr

Abstract

The deployment of large scale grids generates new challenges in system and network monitoring. Users and distributed applications need to access logical and up to date information on sites reachability, links performances and services availability.

In this paper, we propose a presentation model suitable for grid environments, and we present the tool MapCenter developed by The Network Work Package of the European DataGRID that implements this model.

Current grid monitoring technologies are able to store numerous and accurate elementary results in "Grid Information Systems". On a upper level, MapCenter has been designed to logically and graphically represent all elements, applications and services running over grids. It polls grid objects with different methods or make request to Grid Information Systems to build aggregated views of specific grid entities. It displays various views of grids on a Web Site, like graphical maps, logical views of services and full tree of computing elements.

MapCenter offers a flexible presentation layer in huge and heterogeneous environments, as it is in grid contexts. It provides a very simple, autonomous and extensible model that enables the visualization of any level of abstraction (national organizations, virtual organizations, set of data stores, etc) needed by such environments.

1 Introduction

Distributed applications and services spread over large scale and heterogeneous networks imply new paradigms that make traditional monitoring representation models obsolete. More specifically, grid environments [7] [13] presents characteristics that raise the complexity of monitoring and status visualization one step higher.

A grid infrastructure is a complex aggregation of computing, storing and communication physical resources. At an upper level, a grid can be viewed as a aggregation of logical services running on these resources, allowing users and application clients to access and use them.

Grid environments generate new functions and assumptions that must be addressed by a monitoring and visualization tool:

- *Widespread application*: grid application can be running over huge number of sites, spread over several countries and institutes
- *Virtual Organisations*: communities of users (biologists, physicists...) that need to share common views and access rules to the grid.
- *Dynamicity*: fast insertion and removal of computing resources, fast moving of data and files, with dynamic replication systems...
- *Ubiquity*: unknown foreseen localization of running process and end users.

The security of the Grid is also a very important aspect, and monitoring systems have to tackle new constraints to access resources and information:

- All sites in a grid have different security policies (e.g. ICMP packets can be filtered) that raise down the monitoring capabilities
- SNMP [5] access are generally not allowed from outside of site, or it is allowed with restricted access managed with ACL, which is not scalable in grid environments.
- Direct access to grid nodes (e.g. SSH) is not possible in grid environment

Classical monitoring tools over distributed environments have numerous monitoring features (NetSaint [9] , MRTG [15] , BULL OpenMaster [16] , CA Unicenter [17] , IBM Tivoli [18] , HP Openview [19] , ...), but they don't have enhanced functions to fulfil specific grid information presentation and visualization needs.

In this document, we propose a presentation model suitable for grid environments, and we present the tool MapCenter developed by The Network Work Package of the European DataGRID that implements this model.

The main purpose of our model is to response to human's needs: *Final users* to have real-time status of services availability, and *Grid and site administrators* to react quickly and tackle problems occurring in the Grid.

Grid Information Systems [3] (like MDS [11] of Globus [6]) collect and store information from grid resources, and MapCenter is focused on the presentation layer on top of these GIS and grid elements. Figure 1 is an overview of MapCenter positioning compared with other grid entities.

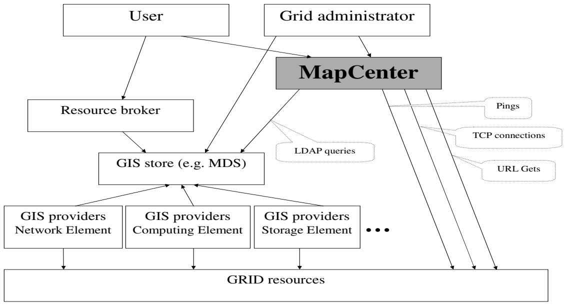


Figure 1: MapCenter overview

In the section 2 we describe the design of the tool, section 3 shows main features and section 4 reports on experimentation and deployment of the tools in the European DataGRID project.

2 Design principles

2.1 Architecture

MapCenter internal architecture (figure 2) is composed of three layers:

- The **Monitoring Layer** polls grid resources through sensors, and Grid Information Systems (GIS) as MDS or GMA, ...
- The **MapCenter Data Store** keeps all internal structures
- The **Presentation Layer** generates and visualize different views of the grid

This structure gives the tool flexibility and extensibility needed; new functionalities like specific monitoring method can be easily integrated.

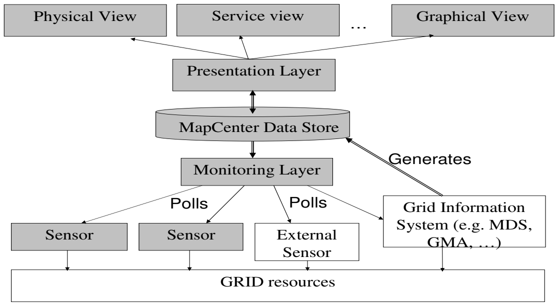


Figure 2: Core structure of MapCenter Processing

2.2 Data Model

Like most of popular commercial monitoring systems, the model uses hierarchical views of resources status, with propagation of monitoring status up to the top of the tree view. The specificity and the strength of our model is its flexibility to generate any logical and graphical views composed of any aggregated parts of the full tree.

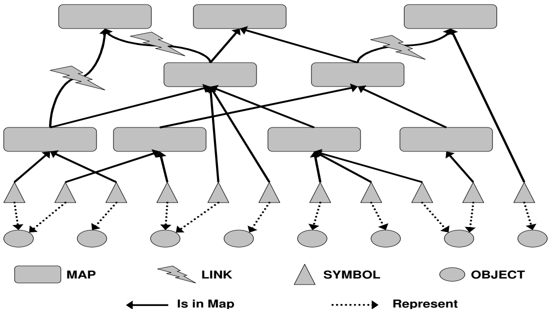


Figure 3: The presentation model of MapCenter

Our presentation model is based on four entities:

- **Object**: An object is the basic element; it generally models a computer or a set of services on a computer which are polled at a fixed frequency. Multiple objects can be defined for the same computer to distinguish different services. Only objects are polled, all other elements (maps, links and symbols) are logical or graphical representation based at the lower level on objects.

- **Symbol**: A symbol is a visual representation of an object; symbols are included and grouped into Maps. Multiple symbols of the same object can be presented in different maps; the corresponding object is polled only once.

- **Map**: a map contains sub-maps, symbols and links. Maps allow the construction of hierarchal views.

- **Link**: A link is an abstraction of a logical interconnexion between maps. It is actually a logical link to a map, with same status. Links are used to have multiple representation of a map in others maps.

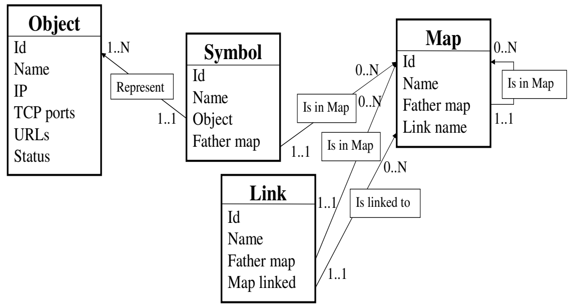


Figure 4: Relational model between the four entities of the presentation layer

A link to a map is a very powerful facility, which enables any representation of logical or graphical maps of complex services, without any duplicated declaration of symbols or maps.

3 Functional model

In this section, we present main features, following the tree levels of the architecture (see figure 2): Monitoring layer, MapCenter Data Store and Presentation layer. We first define the basic components of the design.

3.1 Basic components

As numerous objects have to be polled in parallel, program is multi-threaded. Each thread is represented by a grey box in figure 5.

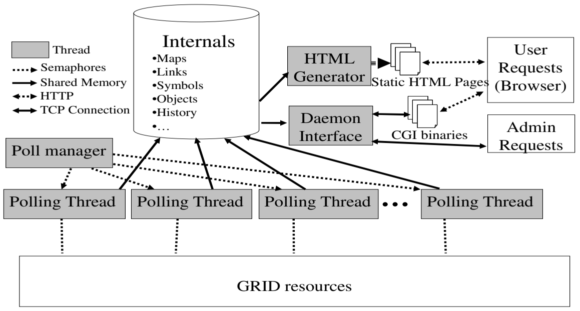


Figure 5: Core structure of MapCenter Processing

- **Polling threads** poll objects with different requests. The number of polling threads is configured (20 threads by default).

- **Poll Manager** distributes the work and schedules polling threads, respecting the configured polling frequency of each object.

- **Daemon Interface** receives requests directly from administrators, or from CGI scripts launched by users from the Web browsing.

- **HTML Generator** generates all static HTML pages as fixed rate (10 minutes by default)

3.2 Monitoring layer

Polling objects

At the lower level, **Polling threads** are monitoring objects. Three types of polling requests are possible:

- *Ping requests*: check the reachability of a machine

- *TCP connections requests* on selected ports: check services running on that machine (e.g. port 80 for HTTP server, port 389 for LDAP ...)

- *HTTP get requests* on selected URLs: get first 128 bytes of one page through internet (or intranet) to check Web services running on that machine

Any number of ports and URLs can be defined for one object.

Status determination rules

Each object is regularly polled with its own frequency, defined in seconds and after each poll, a severity is assigned to the object. After polling, rules are applied to objects in order to determinate the global status of each object. Different rules corresponding to different objects and types of requests have been defined; the principle is to aggregate status information:

```
if (Object is pingable)
  if (Ping failed)
   severity Major
  else if (one TCP failed)
   severity Minor
  else
   if (one URL failed)
    severity Minor
   else
    severity Normal
```

3.3 MapCenter Data Store

The flexibility to represent various comprehensive levels of abstraction of grid resources is implemented by the "MapCenter Data Store", which keeps:

- all configuration information for all elements (objects, symbols, maps, link, services, urls ...).

- status for all these previous elements.

The "MapCenter Data Store" is firstly generated from a single file contains all these data; this is a flat text file which can be easily and quickly modified with any text editor to change configuration.

This text file can also be automatically generated from Grid Information Systems, or by various techniques of dynamic resources discovery (see section "Perspectives").

3.4 Presentation Layer

In parallel with the monitoring polling, **HTML Generator** dynamically generates static pages to represent different views of the status of the grid. Pages are generated at a fixed rate (configurable), and each generated page contains a timestamp of last generation. These pages contains a large number of hyperlinks between themselves, in order to facilitate the navigation.

The core tree view

To represent the status of grid services, the core structure is organised as a tree view (the Full View) that contains all elements of the grid organized by different ways. The tree is composed of maps and symbols. There is no limitation of the depth of the tree view.

Status representation and propagation

Symbols are animated by the corresponding object status, and then all status are propagated along the tree structure:

- Each Map has a *minimum status*: the first icon, which is the best status of symbols and sub-maps in this map.

- Each Map has a *maximum status*: the second icon which is the worst status of symbols and sub-maps in this map, moderated with the maximum propagation parameter of each.

Maximum propagation (mp) parameter controls the maximum status that is propagated to the upper level, which enable to stop propagation of errors that are not important for upper levels (figure 6).

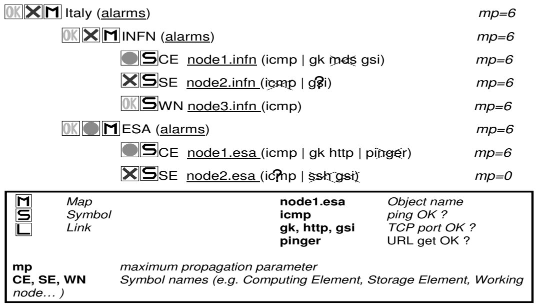

Italy (alarms) mp=6
INFN (alarms) mp=6
CE node1.infn (icmp | gk ~~mds~~ gsi) mp=6
SE node2.infn (~~icmp~~ | g?i) mp=6
WN node3.infn (icmp) mp=6
ESA (alarms) mp=6
CE node1.esa (icmp | gk http | ~~pinger~~) mp=6
SE node2.esa (ic?mp | ~~ssh gsi~~) mp=0
M Map
S Symbol
L Link
node1.esa Object name
icmp ping OK ?
gk, http, gsi TCP port OK ?
pinger URL get OK ?
mp maximum propagation parameter
CE, SE, WN Symbol names (e.g. Computing Element, Storage Element, Working node...)

Figure 6: Example of status propagation

Logical views

The goal is to give back to users the representation of abstract status they need. Logical views are different levels of abstraction of the grid resources and services, that enable to reply to questions like:

- Where a service is deployed and available ?

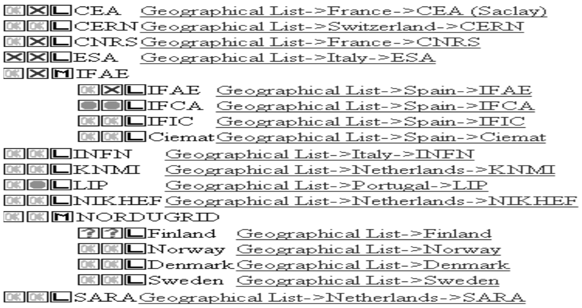
- Where an application can run ?

- What is the status of the resources of a specific country ?

Figure 7 shows for example a logical view of the status of grid ressources offered by organisations participating to the European DataGRID testbed.

From the core tree view, specific parts are extracted, and appear in logical views. Logical views are actually selected parts of the core tree view, and are mainly using links to other maps in this core tree view.

This model is very extensible, and enable to create as many logical views as we can imagine.



- OK ☒ L CEA Geographical List->France->CEA (Saclay)
- OK OK L CERN Geographical List->Switzerland->CERN
- OK ☒ L CNRS Geographical List->France->CNRS
- ☒ ☒ L ESA Geographical List->Italy->ESA
- OK ☒ M IFAE
  - OK ☒ L IFAE Geographical List->Spain->IFAE
  - ● ● L IFCA Geographical List->Spain->IFCA
  - OK OK L IFIC Geographical List->Spain->IFIC
  - OK OK L Ciemat Geographical List->Spain->Ciemat
- OK OK L INFN Geographical List->Italy->INFN
- OK OK L KNMI Geographical List->Netherlands->KNMI
- OK ● L LIP Geographical List->Portugal->LIP
- OK OK L NIKHEF Geographical List->Netherlands->NIKHEF
- OK OK M NORDUGRID
  - ? ? L Finland Geographical List->Finland
  - OK OK L Norway Geographical List->Norway
  - OK OK L Denmark Geographical List->Denmark
  - OK OK L Sweden Geographical List->Sweden
- OK OK L SARA Geographical List->Netherlands->SARA

Figure 7: Logical View of Organisations

Graphical views

Graphical views allow to draw graphics and maps. The goal is to give users a graphical representation of the status of a set of entities. For example, figure 8 shows a graphical view of the European DataGRID testbed sites.

Each maps and symbols in graphical views contains localization information, and a background image. Each map, link or symbol of a graphical view are drawn in a graphic HTML page from these information, with same status determination rules.

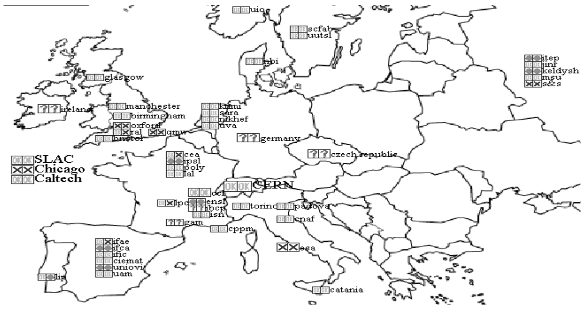
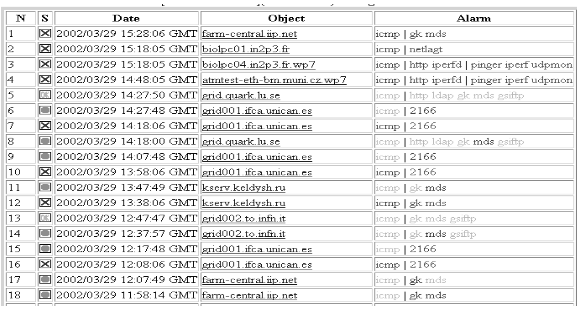


Figure 8: Graphical View of European DataGRID testbed

History

Users need to obtain historical information about the status of services. In order to give such information, an internal event is generated each time the status of an object changes.

The history of last time stamped events is maintained and stored in a rotating buffer. The length of the buffer is defined in a configuration file (1000 events by default). History (figure 9) is displayed in a specific HTML page.



N	S	Date	Object	Alarm
1	☒	2002/03/29 15:28:06 GMT	farm-central.iip.net	icmp | gk mds
2	☒	2002/03/29 15:18:05 GMT	biolpc01.in2p3.fr	icmp | netlagt
3	☒	2002/03/29 15:18:05 GMT	biolpc04.in2p3.fr.wp7	icmp | http iperfd | pinger iperf udpmon
4	☒	2002/03/29 14:48:05 GMT	atmtest-eth-bm.muni.cz.wp7	icmp | http iperfd | pinger iperf udpmon
5	OK	2002/03/29 14:27:50 GMT	grid.quark.lu.se	icmp | http ldap gk mds gsiftp
6	●	2002/03/29 14:27:48 GMT	grid001.ifca.unican.es	icmp | 2166
7	☒	2002/03/29 14:18:06 GMT	grid001.ifca.unican.es	icmp | 2166
8	●	2002/03/29 14:18:00 GMT	grid.quark.lu.se	icmp | http ldap gk mds gsiftp
9	●	2002/03/29 14:07:48 GMT	grid001.ifca.unican.es	icmp | 2166
10	☒	2002/03/29 13:58:06 GMT	grid001.ifca.unican.es	icmp | 2166
11	●	2002/03/29 13:47:49 GMT	kserv.keldysh.ru	icmp | gk mds
12	☒	2002/03/29 13:38:06 GMT	kserv.keldysh.ru	icmp | gk mds
13	OK	2002/03/29 12:47:47 GMT	grid002.to.infn.it	icmp | gk mds gsiftp
14	●	2002/03/29 12:37:57 GMT	grid002.to.infn.it	icmp | gk mds gsiftp
15	●	2002/03/29 12:17:48 GMT	grid001.ifca.unican.es	icmp | 2166
16	☒	2002/03/29 12:08:06 GMT	grid001.ifca.unican.es	icmp | 2166
17	●	2002/03/29 12:07:49 GMT	farm-central.iip.net	icmp | gk mds
18	●	2002/03/29 11:58:14 GMT	farm-central.iip.net	icmp | gk mds

Figure 9: History

Dynamic behaviour

To improve interaction in problem detection, CGI scripts have been developed to generate dynamic pages. The user can for example view alarms of a specific object or map, or ask for a ping to a machine, or even ask for a service scan of a machine.

3.5 Information Systems Queries

Users can directly query Information Systems of grids. A LDAP [8] browser CGI script has been included that enables to ask for any information stores in any LDAP hierarchy. As an example, users have direct access to all MDS information of Globus from MapCenter.

Access to any other Information System can be developed. Scripts for accessing R-GMA (a DataGRID Relational GMA implementation) will be included.

4 Experiments and results

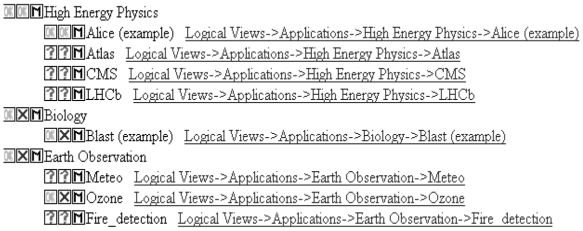
MapCenter has been developed and deployed within the European DataGRID project [2]. This project started in 2001 and will continue until the end of 2003. The DataGRID testbed will be totally used as a production grid for physicists, biologists, and earth observation sciences after 2003. DataGRID interconnects forty sites over Europe, and MapCenter is daily used to view the status of all services running on all these sites, and to track the deployment of successive versions of DataGRID middleware and applications [1].

The current testbed is composed of:

- 7 applications distributed among 3 virtual organisations (figure 10)

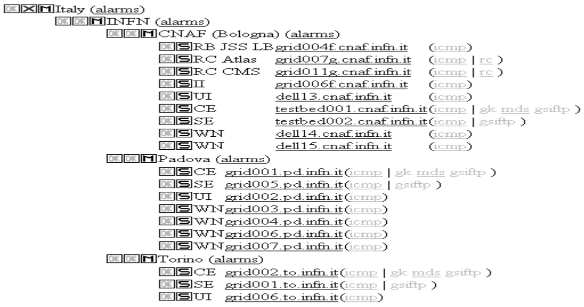
[1] Please visit web site [1] to have a more complete view of MapCenter results.

- 11 organisations over 15 countries (figure 7)
- 40 sites in Europe (figure 8)
- Large diversity of grid elements on each site (figure 11)



```
OK OK M High Energy Physics
        OK OK M Alice (example)  Logical Views->Applications->High Energy Physics->Alice (example)
        ? ? M Atlas  Logical Views->Applications->High Energy Physics->Atlas
        ? ? M CMS  Logical Views->Applications->High Energy Physics->CMS
        ? ? M LHCb  Logical Views->Applications->High Energy Physics->LHCb
OK X M Biology
        OK X M Blast (example)  Logical Views->Applications->Biology->Blast (example)
OK X M Earth Observation
        ? ? M Meteo  Logical Views->Applications->Earth Observation->Meteo
        OK X M Ozone  Logical Views->Applications->Earth Observation->Ozone
        ? ? M Fire_detection  Logical Views->Applications->Earth Observation->Fire_detection
```

Figure 10: Logical View of Applications



```
OK X M Italy (alarms)
    OK OK M INFN (alarms)
        OK OK M CNAF (Bologna) (alarms)
            OK S RB JSS LB grid004f.cnaf.infn.it       (icmp)
            OK S RC Atlas  grid007g.cnaf.infn.it       (icmp | rc )
            OK S RC CMS    grid011g.cnaf.infn.it       (icmp | rc )
            OK S II        grid006f.cnaf.infn.it       (icmp)
            OK S UI        dell13.cnaf.infn.it         (icmp)
            OK S CE        testbed001.cnaf.infn.it(icmp | gk mds gsiftp )
            OK S SE        testbed002.cnaf.infn.it(icmp | gsiftp )
            OK S WN        dell14.cnaf.infn.it         (icmp)
            OK S WN        dell15.cnaf.infn.it         (icmp)
        OK OK M Padova (alarms)
            OK S CE  grid001.pd.infn.it(icmp | gk mds gsiftp )
            OK S SE  grid005.pd.infn.it(icmp | gsiftp )
            OK S UI  grid002.pd.infn.it(icmp)
            OK S WN grid003.pd.infn.it(icmp)
            OK S WN grid004.pd.infn.it(icmp)
            OK S WN grid006.pd.infn.it(icmp)
            OK S WN grid007.pd.infn.it(icmp)
        OK OK M Torino (alarms)
            OK S CE  grid002.to.infn.it(icmp | gk mds gsiftp )
            OK S SE  grid001.to.infn.it(icmp | gsiftp )
            OK S UI  grid006.to.infn.it(icmp)
```

Figure 11: Part of the Core Tree View

MapCenter has been developed to be independent from any grid and system, and it can be deployed on any distributed environment. It is currently also deployed over RNTL E-toile project, a new French grid testbed.

5 Perspectives

5.1 Dynamic resources discovery

All information (objects, maps, logical views...) can be configured by hand, but huge grid environments impose the challenge of automate monitoring and visualising configuration to respond to scaling issue.

In order to achieve a first step in this configuration process, we are developing scripts to dynamically generate "Data Store" from Grid Information System (GIS).

We currently have scripts to query GIS of Globus middleware (MDS), and we are able to automatically generate following information:

- **Resources Discovery**: from MDS, we get all objects on all sites for all organisations, and we are able to build the full tree view of physical resources running on the grid.
- **Logical Discovery**: from MDS, we can also discover Virtual Organisations distribution, and know which application is running on which grid elements. We generate automatic logical views from this information.
- **Location Discovery**: from localisation information of each grid elements (Whois [21], NetGeo [20], ...) , we are able to generate graphicals views of grid services.

These scripts are supposed to run on a regular basis (e.g. each night) and they will generate the main part of "Data Store". The solution we have chosen is to get most of grid configuration automatically, but also to let administrators add "human minded" configuration if they want.

In a second step, we are working on getting more information from other GIS and we are also studying dynamic discovery agent and methodologies [14].

5.2 The Future of MapCenter

Many functionalities can be added easily. We are currently working on the following ways:

- **Presentation Layer**: add sub-logical views in logical views, and add graphical sub-maps in graphical maps.
- **Monitoring Layer**: add new sensors [4] (e.g. query of SNMP agents, query of information from GIS...)
- **Security**: get restricted access to the web site, grant access to main daemon (e.g. access to LDAPS with X.509 certificates) and integrate MapCenter and GSI [12] (e.g. check "GSI connection" instead of TCP connection)
- **Distributed Architecture**: add supra-management capabilities beetwen MapCenter servers. For example, in DataGRID, we could have one server server per country, and a top-level one.

- **Integration with OGSA [10]**: The flexibility of MapCenter will allow an easy integration with future Open Grid Services Architecture implementations.

6 Conclusion

MapCenter is focused on the presentation layer in a huge and heterogeneous environment, like a grid context.

It is offering very flexible and simple model that enables representation of any level of abstraction (national and international organizations, virtual organizations, applications) needed by such environments. Implementation is totally autonomous (standalone system independent from grid middlewares), extensible and it can interoperate with any Grid Information Systems.

MapCenter is currently running to monitor European DataGRID [1]. Development is ongoing and new functionalities will be added and it will be deployed soon over E-toile, the French Grid Project.

References

[1] MapCenter Web Site for DataGRID
http://ccwp7.in2p3.fr/mapcenter

[2] European DataGRID projet
http://web.datagrid.cnr.it

[3] B. Tierney, R. Aydt, D. Gunter, W. Smith, V. Taylor, R. Wolski, M. Swany, "A Grid Monitoring Service Architecture", Global Grid Forum Performance Working Group, 2001.

[4] B. Tierney, B. Crowley, D. Gunter, M. Holding, J. Lee, M. Thompson, "A Monitoring Sensor Management System for Grid Environments", Proceedings of the IEEE High Performance Distributed Computing conference (HPDC-9), August 2000.

[5] D. Harrington, R. Presuhn, B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", IETF Draft, 16 October 2001.

[6] Globus http://www.globus.org

[7] I. Foster, C. Kesselman, "the Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, Pub.ISBN 1-55860-475-8, August 1998.

[8] Wahl M., T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)", IETF RFC 2251, Dec. 1997.

[9] NetSaint //www.netsaint.org

[10] I. Foster, C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", January, 2002.

[11] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Services for Distributed Resource Sharing", Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[12] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids", Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.

[13] I Foster, C. Kesselman, S Tuecke, "The anatomy of the grid: Enabling scalable virtual organisations", International Journal on Supercomputing Applications, 2001

[14] A. Iamnitchi, I. Foster, "On Fully Decentralized Resource Discovery in Grid Environments", International Workshop on Grid Computing, 2001

[15] MRTG http://www.mrtg.org

[16] BULL OpenMaster http://www.openmaster.com

[17] CA Unicenter http://www.cai.com/unicenter/

[18] IBM Tivoli http://www.tivoli.com/

[19] HP OpenView http://www.openview.hp.com/

[20] NetGeo - The Internet Geographic Database http://www.caida.org/tools/utilities/netgeo/

[21] Whois.net http://www.whois.net/