A Web Interface Toolkit to Manage the AMGA Metadata Server Within the EGEE Standard Deployment

Víctor Méndez Muñoz, Gabriel Amorós Vicente, Carlos Escobar Ibañez, Mohammed Kaci, Javier Nadal Durà
Grid and e-Science Group.
Instituto de Física Corpuscular (IFIC)
- mixed institute of Consejo Superior de Investigaciones Científicas (CSIC) and Universitat de València (UV) -
Apt. Correus 22085, E-46071 , València, Spain
Corresponding author e-mail: vmendez@ific.uv.es, Other e-mails: {amoros,carlos.escobar,kaci,jvnadal}@ific.uv.es

***Abstract*—This paper describes a Web Graphical User Interface (GUI) toolkit for the access of the scientists to the official Enabling Grid for E-sciencE (EGEE) metadata server (AMGA). Such toolkit is composed of the amgaNavigator program and a library of PHP4 API to the AMGA server. The amgaNavigator is a high-level Grid interface designed for the current EGEE-III stable operating system (Scientific Linux 4) and the public package distribution. The contribution to the Computer Science is a toolkit, which integrates all the necessary deployment stacked on the EGEE standard platform, keeping the platform homogeneity with the rest of the infrastructure services. For the users, the amgaNavigator offers an exploration of the metadata schema and entries, with advanced searches in the catalog, avoiding the end-user to handle the AMGA SQL syntax.**

***Keywords*-Grid Computing; Information Interfaces; Software Engineering; Software Architecture; Reusable Software**

I. INTRODUCTION

The metadata service allows transparent access to information stored in distributed resources. Any metadata service for scientific purpose of different potential end-users and applications involving groups of many research centres, must integrates them on a standard and collaborative information system. For this reason, the gLite, which is the EGEE middleware [1], has adopted the AMGA as the metadata service [2]. The integration in a Grid infrastructure allows the access to the metadata associated to the Virtual Organization (VO), which the user belongs to. The permissions and flexibility of the account management that offers the VO context, is translated to AMGA smoothly. The issues of security, privacy and other related to AMGA, are explained in other works [3,4]. One clear example of AMGA use is for e-Health where typically hospitals or medical physics research centres, have large amount of information that, by legal issues, are not allowed to be exported outside their buildings. However, very often, the studies or the statistics needed by them, must include data stored in several of these centres. Thus, AMGA offers the possibility to collect metadata about this data and facilitate the search and the access to the data keeping on it the rights and permissions. A concrete example is a scientist involved in a study collaboration with several hospitals, to which is necesary the access to specific information with several keywords. He can use AMGA searching with the keywords and, since he has the correct permissions, AMGA acts as a portal to bring him the data that is returned by the documents found with the search, and only gives access to the data related to the study.

In this scientific context, we have identified some metadata services requirements in our research institute. In addition to the user metadata requirements, we have considered our complex Grid infrastructure context. Since we have deployed two Grid infrastructures in our institute, the platform and middleware standardized deployment is a key issue for efficient administration. The first of these infrastructures is a Tier-2 / Tier-3 of the Large Hadron Collider (LHC) Europe Southwest regional Grid infrastructure [5], supporting Grid services on LHC Computing Grid (LCG) basis, mainly for the ATLAS experiment. The other infrastructure is a Grid-CSIC [6] site running scientific applications of all the CSIC institutes. The Grid-CSIC deployment is keeping homogeneity with the EGEE platform, the package distribution and the middleware standards. An additional reason for a standardized deployment, is the calling to integrate the Grid-CSIC in the future NGI/EGI, which starts to operate following and enhancing the EGEE-III in spring of 2010 [7].

The use of AMGA will provide the metadata features to the Medical Physics Grid environment of our researchers and collaborators [8,9], and potentially other scientific applications with metadata requirements. The amgaNavigator toolkit is designed to improve the metadata accessibility by the user point of view, integrating and connecting other Grid services of our complex Grid infrastructure, and providing a technical solution for the standardized deployment of all the services involved, not only the middleware but also the platform dependent components like the databases or the web servers.

In the access to the AMGA service, instead of the native command line client, our end-users need a GUI. The available AMGA GUIs does not cover our special requirements. For deployment reasons we need a Web client

 ISBN: 978-1-61208-101-4

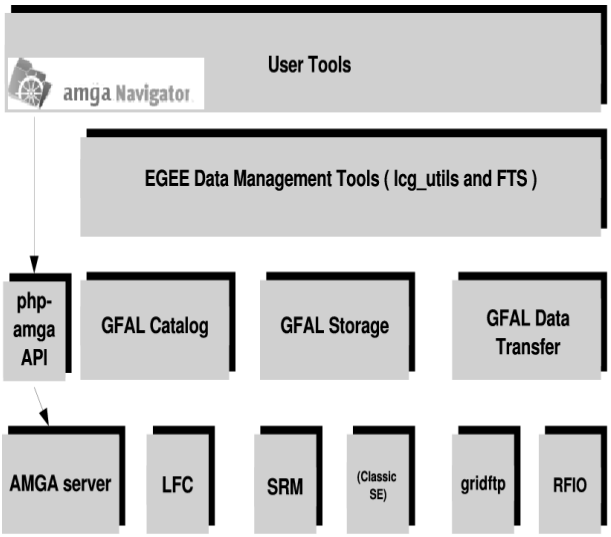

Figure 1. The amgaNavigator framework

accessible for our contributors over the world, without any client installation and support, just the Web browser. For maintenance reasons, the architecture may use the stable EGEE operating system, nowadays the Scientific Linux 4 and the corresponding packages distribution. We have developed a down-porting of AMGA API on PHP5 [10] to the public EGEE stable release on PHP4, and we have built amgaNavigator upon this API following the EGEE standards. The rest of this paper is a related work Section of the catalog components and the metadata user interface. Following, it is described a design Section of the amgaNavigator, which also explains the motivations of this application. We continue with the amgaNavigator detailed functionalities description, and finally conclusions and future works.

II. Related Work

The EGEE architecture is partially web services oriented, mostly in the job scheduling services. Moreover, part of the architecture is component-based, as a result of historical evolution of the gLite middleware. Such components are accessible through wrappers supporting the service oriented functionalities. In Figure 1 it is shown the AMGA service as part of the EGEE architecture. It is a basic grid service used by Grid File Access Library (GFAL) for the catalog management. GFAL is an abstraction of the storage, catalog and transfer specific services. GFAL is used by lcg utilities and high-level transfer services, which furthermore of the high-level functionality it support the VOs context and permissions. The AMGA service is also directly accessible by end-users or applications [11], without mediation of the stacked services.

When metadata access is deployed, two services give the main catalog management functionalities: the LHC File Catalog (LFC), or other catalog services like FiReMan [12], and the AMGA metadata service. AMGA is a metadata catalog organized as a filesystem, where we can find directories with schemas defined by different attributes for each directory. For any Logical File Name (LFN) we will find an entry in the AMGA server, which gives values for the schema attributes. Such values of the attributes are the metadata associated to the file. The AMGA relates the LFN with the Grid Unique Identifier (GUID), but does not provide any information about physical location of the file replicas, which should be supplied by the catalog service.

The AMGA client is a command line shell. This is a rude access to the service, specially when end-users have to interact with the AMGA for common operations. Therefore, some research groups have developed AMGA GUIs for different purposes.

Amga Browser [13] is a Python client for generic access to AMGA server. Amga Browser allows a graphical exploration of the metadata schema, and a command line launcher and text results screen. The commands are on AMGA SQL like, so the end-user must have some knowledge of such syntax.

LHCb Book-keeping Database Browser [13], is a very specific AMGA utility for browsing of the LHCb logging and booking system. It is a Java and Python client integrated in the Ganga Grid UI.

The INFN team have developed the AMGA server and also some Web GUIs integrated on applications designed for specific environments. The first one is the INFN-Catania Web Front-end, a GUI to access all the metadata related to gMOD, the Grid Movie On Demand service of Genius Portal [14]. The AMGA WI, a metadata Web Interface of GILDA [15] project, developed based on P-GRADE [16] with Java technologies using the AMGA Java API [17]. Another example is gLibrary [10] the Grid digital assets management software based on AMGA PHP5 API.

III. Design

There are some reasons that motivate the design and implementation of the new AMGA UI in our complex Grid infrastructure.

- We need an AMGA Web GUI, for an affordable deployment and maintenance of the client side.
- The AMGA Web GUI must have an available open-source distribution to support the application modifications and integration on our scientific platforms, without third-party provider dependencies.
- The AMGA Web GUI server side should run over the stable Scientific Linux, nowadays SL4, and the package software dependencies should be the stable package distribution, which includes PHP4 and MySQL4, for homogeneous administration.
- The AMGA Web GUI software architecture should be reusable for future integration of other services.
- The AMGA Web GUI should be designed to fulfil the end-user requirements.

 ISBN: 978-1-61208-101-4

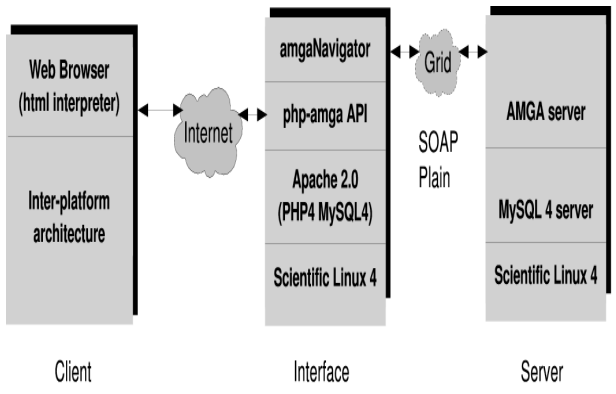

Figure 2. The amgaNavigator client/server

Figure 1 shows the architecture framework of the application. We can see that amgaNavigator offers a high-level middleware service on the user tools layer. The amgaNavigator uses amga-php API for direct access to the AMGA server, regardless of the EGEE stack architecture (GFAL, LCG utilities and file services). On such architecture framework, the amgaNavigator may access other data management services, to integrate or launch user operations, for example, catalog services for replica location and selection or VOMS for the X509 certificate authorization.

In Figure 2, we find the amgaNavigator client/server schema. The client environment can be a platform where a Web browser is installed. The Web browser receives from the amgaNavigator the html tags and embedded information and sends the data forms. In the server side the architecture is stacked as shown in Figure 2. The Apache 2.0 server is configured with PHP4 and MySQL4 modules. PHP provides dynamic Web execution on Apache server. The amgaNavigator uses the php-amga API access to the AMGA server, with MySQL database back-end. The AMGA server offers other back-ends like PostgreSQL. The php-amga communication with AMGA server is possible with SOAP or connection oriented using plain information on TCP sockets. The different servers involved with amgaNavigator: MySQL, AMGA and Apache, can be deployed on a local or remote host.

The design of amgaNavigator has been raised with reusable software techniques in the design of the API library to connect the AMGA server. We need an application that uses the distribution of the stable release EGEE platform. Thus, starting from the official amga-php API, nowadays on PHP5, we have ported to PHP4 and we have included some functionalities, mainly to parse the AMGA strings to fulfil amga-php syntax and to retrieve results from the API. The updated amga-php API on PHP4 would be valid for next PHP releases, due to descent compatibility feature of PHP project. This PHP class comes with License for EGEE Middleware, which basically is an opensource with some restrictions for distribution out of the scientific and academic environments.

We also apply reusable engineering of PHP File Navigator (PFN) [18], with GPL distribution, to reduce the implementation and debugging time. Therefore, the obtained amgaNavigator is also GPL distributed, which increases the possible collaborative work comparing with proprietary licensing restrictions. We have used the CSS Styles, dynamic language support (PFN vars class), configuring functionalities (PFN conf class), icon and imaging management (PFN imx class) and we did re-engineering with some parts of the PHP presentation templates.

IV. FUNCTIONALITIES

We have considered the common AMGA user operations and we have implemented them in the amgaNavigator:

- Control access
- Browse the AMGA directories and entries structure
- Create directories and assign attributes to them
- View and modify directories attributes
- Delete directories or their attributes
- Create entries and assign attribute values to them
- View and modify the attributes entries values
- Delete entries
- Advanced search by multiple conditions
- View and modify the file permissions
- Multiple delete operations
- Multiple permissions assignment operations.

AmgaNavigator offers dynamic language support. The main components of the web homepage may be summarize as following. The header has the main menu, wich has some icons to update information of actual directory browsing, create directory form and create entry form. The top right main menu offers links to advanced search and exit. On the body of the web page, the user can explore into directories getting their contents. Each line of the list is related to a directory or entry information: name, type, owner, permissions, and associated actions for the individual entry or directory (edit, permissions, delete). The list has check boxes for multiple operations, launched on the bottom of the list. The possible multiple actions are delete directories or entries and set permissions.

The rest of the web pages, reports and forms, are composed with the main components and other specific functionalities components as well. An example of advanced search is shown in Figure 3. Regardless of the mentioned main graphical components, the specific components allow the user to select any attribute from a pull-down list to define the searching conditions. Figure 3 presents the result of the attributes condition search for *hubble<0.7 and omega<0.6* in the directory */Inicio/planck/cosmology*. If more conditions want to be introduced, then it has to be clicked the *and/or* buttons. Below those buttons we can see that no pattern has been introduced as condition of the entrie names. Only file *007* satisfied all the search conditions.

 ISBN: 978-1-61208-101-4

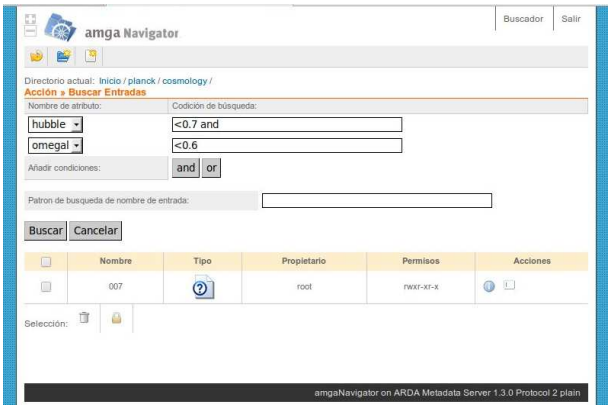

Figure 3. Advanced search screen shot example

The amgaNavigator attends user specifications on metadata common operations, with advanced graphical mechanisms not available in the AMGA server, like fast operations on repetitive task when is needed, for example, adding entries. Other improvements on the accessibility to the native service are the multiple operations with single step for selected items; the modify operations integrated with the corresponding display values; or the visual icon information and links to browse. The users have on a single web page all the necessary information for any operation of their common metadata requirements, with improved usability over AMGA server.

V. Conclusions and Future Work

We have developed a browser based on probed code components. It has been developed with an easy to maintain criteria, following the POSIX code presentation and the PHP object oriented scalable code structure. By the user point of view amgaNavigator offers an interface integrated with the EGEE data management to improve the accessibility of the AMGA server, providing additional functionalities. All the necessary deployment is also possible following the EGEE standards, which is an important computer administration issue. For this purpose we have developed a down-porting of the amga-php API, to the stable PHP4 package on SL4 release. In this manner we keep more homogeneity within the complex Grid infrastructure.

The software architecture is designed to integrate future requirements of other EGEE data services. Additional future work is the implementation of advanced user features, like schema copies, or functions not contemplated on AMGA server, like the recursive search. When our Medical Physics Grid environment will reach a production state, a critical mass of users will provide the feedback of amgaNavigator accessibility and aditional services integration requirements.

Acknowledgements

We are greatly indebted to the founding agency Spanish National Research Council-CSIC for their support from project Grid-CSIC.

References

[1] M. Lamanna, M.-E. Begin, Z. Sekera, and S. Collings, “Egee middleware architecture,” CERN (EGEE-DJRA1.1-594698-v1.0), Tech. Rep., 2005.

[2] M. Schulz, “glite data management and information system,” in *EGEE User Forum*. OGF20/EGEE, 2007.

[3] J. Montagnat, D. Jouvenot, C. Pera, A. Frohner, P. Kunszt, B. Koblitz, N. Santos, and C. Loomis, “Bridging clinical information systems and grid middleware: a medical data manager,” *Studies in health technology and informatics*, vol. 120, pp. 14–24, 2006.

[4] J. Montagnat, A. Frohner, D. Jouvenot, C. Pera, P. Kunszt, B. Koblitz, N. Santos, C. Loomis, R. Texier, D. Lingrand, P. Guio, R. D. Rocha, A. S. de Almeida, and Z. Farkas, “A secure grid medical data manager inteface to glite middleware,” *Journal of Grid Computing*, vol. 6, pp. 45–59, 2007.

[5] S. G. de la Hoz, L. March, E. Ros, J. Sánchez, G. Amorós, F. Fassi, A. Fernández, M. Kaci, and A. Lamas, “Analysis facility infrastructure (tier-3) for atlas experiment,” *The European Physical Journal C*, vol. 54, pp. 691–697, 2008.

[6] Grid-CSIC-Project, “Grid infrastructure for advanced research at the spanish national research council (grid-csic),” 2010, http://www.grid.csic.es/.

[7] WP5, “Egi blueprint: Design study,” CERN, Tech. Rep., 2008, http://web.eu-egi.eu/fileadmin/public/EGI_DS_D5_3_V300b.pdf.

[8] G. Amorós, F. Albiol, J. Ors, A. Fernández, S. González, M. Kaci, A. Lamas, L. March, E. Oliver, J. Salt, J. Sánchez, M. Villaplana, and R. Vives, “Scientific applications running at IFIC using the grid technologies within the s-science framework,” in *Third International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*. IEEE, 2009.

[9] Partner-Project, “Particle training network for european radiotherapy,” 2010, http://cern.ch/partner.

[10] A. Calanducci, C. Cherubino, L. N. Ciuffo, and D. Scardaci, “glibrary: Digital asset management system for the grid,” in *Conference on Hypermedia And Grid Systems*. IEEE, 2007.

[11] N. Santos and B. Koblitz, “Distributed metadata with the amga metadata catalog,” in *Workshop on Next-Generation Distributed Data Management (HPDC-15)*. ACM, 2006.

[12] C. Munro, B. Koblitz, N. Santos, and A. Khan, “Performance comparison of the lcg2 and glite file catalogues,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 559, no. 1, pp. 48 – 52, 2006.

[13] D.Piparo, “Two graphical browsers for the amga metadata cataloge,” in *Nuclear Physics B, Proceedings Supplements*. Elsevier B.V., 2007, pp. 311–313.

[14] Genius-Portal, “Grid enabled web environment for site independent user job submission,” 2009, https://glite-tutor.ct.infn.it.

[15] GILDA-Portal, “Grid INFN laboratory for dissemination activities,” 2010, https://gilda.ct.infn.it/testbed.html.

[16] P-Grade-Portal, “Parallel grid run-time and application development environment,” 2010, http://www.p-grade.hu.

[17] S. Scifo and V. Milazzo, “Amga wi - amga web interface,” in *Conference on Hypermedia And Grid Systems*. IEEE, 2007.

[18] PFN-Project, “Php file navigator,” 2010, http://pfn.sourceforge.net.