Grids Portals: Frameworks, Middleware or Toolkit

Xavier Medianero-Pasco[1], Belén Bonilla-Morales[2], Miguel Vargas-Lombardo[3]

**[1,2,3] Technological University of Panama
Panama City, Panama**

Abstract

Grid portals are interfaces for interconnection between Grid resources and users. Grid portals are supported by components called portlets, servlets and middleware. Frameworks, middleware and/or development toolkits are used in the design and development of grid portals; each of them presents distinctive characteristics like any existing tool for grid portals such as: GridSphere, GridPort, PortalLab and others. The concept and definition of the components of a Grid Portal are not clear at the time of their application, because different models and works are appointed arbitrarily without taking into account the approach and features of the concept. Their clarification can give an insight to each grid portal and existing components.

Keywords: *Grid Portals, Middleware, Portlets, Servlets, Framework.*

1. Introduction

Grid computing is a technology that enables coordinated exchange and use of hardware resources, software and information in virtual organizations [1]. These resources are not centralized, but distributed extensively across the web. Grid computing environments are dynamic and open and have autonomy to control services.

Generally, grid portals give access to the Grid structures through portals or access points [2]. Grid portals are web-based user interfaces which act as interfaces relating environments to users within the Grid environment. The main function of a grid portal is to interconnect different grids, in other words, to unify and compact the information search and utilization of hardware resources and software, and to hide the level of complexity of the Grid, enabling inexperienced users to access and use the Grid.

Grid portals emerged as a solution for the need of a simple and straightforward environment for grid exchange, so that the middleware can handle distributed resources provided by the user through the portal.

The next generations of Grid portals are called semantic Grid portals [3], whose objective is to integrate content, data tracking and semantic information search. A semantic portal is responsible for searching the meaning of the functions by itself: describe what it does, give details of its uses and the information generated, thus increasing the impact rather than simply establish which variables need and in response returns.

There are several projects involved in the development of Grid portals, some of them related to semantic portals [3] such as: Portals Finland Museum, Cultural Relic Museum, Onto Web Portals, UDM Grid, among others. The community-oriented generation of Grid portals have design projects such as: Portlab, Grid Portal Development Kit, Astrolable, Gridsphere, GridPort Toolkit, among others [3,4,5]. All these projects have a common objective: to interconnect different points (Clusters and Grid) in a larger Grid.

The components of Grid portals are designed portal Frameworks, Middleware, a set of Java components, Servlets, Portlets, including technical differences depending on architecture, structure, functions and components.

This paper consists of several sections. Section II defines Portals and states their characteristics, definitions and models. Section III defines concepts, structure and elements of Frameworks, Middleware and components. Section IV will provide information about Portlets and Servlets. Section V provides the proper classification of some Web portals whose definition does not match the correct concept. Section VI presents the conclusion of research.

2. Portals

Portals are Web applications that enable easy user interaction with services and resources of the grid [6]; they utilize a support infrastructure called: portlet, which can be implemented in conjunction with the servlet.

A Grid portal is an application server that offers an online environment, insurance for collecting information on services and resources of the network [7]. In addition, the Grid portal is the access point to a web system; it provides an environment where the user can access the resources and services of the Grid, implement and monitor network applications and collaborate with other users [3]. Grid Portals are responsible for managing the entry and authentication of users (which play roles as authors, searchers, administrators, researchers) through an interface that provides a list of information resources, job scheduling and file transfer [8].

The main disadvantage of websites is that once they are configured it is very difficult to adapt a new application [2].

Grid portals are made up of portlets that are responsible for incorporating the same features including validations and search engines across the grid. Each service represents a portlet so that the removal, modification or addition of Web services results in a small impact, since the cohesion between services is minimal.

Portals can be classified into two models depending on their execution form, and ubiquitous access (present everywhere). OGCE and HotPage are examples of portals that provide ubiquitous access; the user handles certain aspects of the Grid portal including accounts with high performance computing resources, have setup and configuration responsibilities, and others. Portals that are controlled directly by the administrator correspond to the second model. An example of the second model is Punch In-Vigo, where the administrator is responsible for almost all manufactured Grid portal [5].

According to [3], the portals can be classified into two models depending on user control.

Some portals provide access to high-performance computing resources at anytime; users have control access to any resource to install what they need. Note that in this type of portal users have knowledge of the Grid middleware itself, and therefore are required to implement changes.

Other portals provide access and resource control to a specific user administrator who is responsible for installing, configuring, or running applications and middleware. Users (clients) only interact with the portal and are unaware of the procedures running under it.

Of the two previous models, the second provides a more stable and controlled setting, because the control of the portal is the responsibility of the administrators and users only perform the operations they need, knowing that the level of knowledge of the administrator, a specialized user, is much higher than that of the users although they know the Grid works partially in the first model. These two models have advantages and disadvantages; therefore, the model that best suits the needs of what wants to be deployed is the one that should be used.

According to [6], portals can be classified into:
Portals targeting users: for a given area of a population.
Portals oriented applications: for specific applications and uses of the Grid.
Portals Oriented Development: tools that are portals for the development of new portals.

2.1 Grid Portal based on GPIR [7]

The GPIR (Grid Port Information Repository) is a framework that has three layers: user, service and application. The grid architecture underlying GPIR have three layers: resource layer, data layer and client tier. The resource layer controls and integrates the resources of the grid; the database layer keeps track of resources you have, while the client tier allows user interaction with the Grid portal.

Concerning the framework, the user layer is responsible for the interaction of the customer with the Grid portal, specifically search and user tools. The service layer controls authentication, security modules, and Grid services that are not linked directly to the users. Meanwhile the resource layer is responsible for monitoring, storaging and handling Grid-related information under a specific portal.

2.2 UDM Grid [3]

University Digital Museum (UDM) is a Chinese Information Grid that aims to integrate resources of some thirty museums scattered across the network to share resources, filter and sort information according to the requirements and knowledge level of the users, and eliminate data islands connecting all subnets.

The UDM is an application of Grid MIPGRID (Massive Information Processing Grid) which was designed in 2004 to solve the problems of integration and heterogeneity of information across the various virtual museums in China [9].

There are several projects carried out on UDM Grid, which were studies on the progress of the portal, tools for the design of ontology and semantic annotations on the UDM Grid [9], both in 2006. This project is currently run by China Education Research groups.

The UDM serves as a virtual digital museum that facilitates access to exchange information, or utilization of resources. It has incorporated a Grid portal which consists of several layers of services which are as follows:

Grid Resource Layer: represents the scattered resources that must be added and integrated into the Grid.

Grid Service Layer: stores unified layer Grid resources services allowing access to the portlet services. This layer is the ontology of the system's service.

Portlet Service Layer: a layer with the function of setting logic to applications and services and to adjust the Grid hiding through the portal the complexity of the Grid to the users.

Portlet Layer: contains the portlets used as facilities for inserting the Grid portal that provides ontology services.

Application Layer: contains the portlets stored in the portlet layer and deploy it in the application that must be deployed and run. Application services are indifferent to the user.

User Layer: is the user of the portal including standard users, resources and suppliers.

These layers are integrated to form the UDM Grid architecture; users interact with the application layer where elements are deployed portlet layers, which are run and controlled by service portlets that are responsible for linking resources to the Grid.

The UDM Grid architecture consists of twelve modules, which are responsible for the following functions:

Semantic search (meaning methods and functions, what it does, how they are applied and y are processed), converts a query to data schema, determines ontology, provides access to heterogeneous databases, details, classification of the museum information resources, database and semantic annotation tools. It integrates heterogeneous resources to the information network including information, knowledge and software resources, stores the results of user requests in the portal to increase efficiency when the same resources are accessed again, and integrates resources obtained from database access services and Web service resources.

2.3 Grid Sphere [11]

Grid Sphere is a framework of a Grid Web portal that allows the integration of portlet and servlet containers. It enables the design of portals in a fast and structured way.

Grid Sphere has a set of Portlets that allows you to perform the following functions:

Authentication and Log services, application and maintenance of accounts, user profile and maintenance, layer configuration, portlet subscription , managing local files, notes and instant messaging.

3. Frameworks, middleware, toolkits?

There are differences between the concepts of Framework, Components and Middleware:

3.1 Frameworks

Frameworks are designed structures that contain modules, methods and software features whose function is to be a template for applications derived from the same frameworks. In other words, using the same core and are covered by a specific area.

Conceptually, frameworks are composed of two sections: the core, which represent classes, libraries, methods and modules that are equal and that serve as the basis for all applications, thus, , all applications generated by this framework will have the same basic features stored in this layer. The second section is the slots which represent the elements that can be adapted, added, or simply ignored in the application. You can imagine slots as checklists where you choose additional methods that the application must have and the elements in the slots that are not required for the framework to function. .

Characteristics of the frameworks:

Frozen spots: represent those points of the framework that are neither extensible nor adaptable. Frozen spots are the basic components of the framework.

Hot Spots: are highly extensible, adaptable and configurable.

Extension Points: are the segments that might link hot and frozen spots, or two hot spots.

3.2 Middleware

Middleware is connectivity software that serves as intermediary between the various platforms in which the portals can be. Middleware interacts with physical resources such as users and built applications [10].

The middleware emerged in the 1980's aiming to integrate applications that were on different platforms and / or operating systems.

Characteristics of middleware:
Middleware are independent of the applications, use standard communication protocols, and are run independently to the platform.

3.2.1 In-Vigo [5, 10]

In-Vigo is a middleware that provides a distributed environment where multiple applications coexist in virtual and physical resources and it uses virtualization to create virtual resources.

It was developed by ACIS Laboratory using Java Servlets as the core element of the system which was first implemented in 2004. It has different applications, is used in DineroIV (cache simulator for memory), can be configured to run on a large number of trial and scientific applications such as XSpim (Simulator) and Gamma (a chemical package testing).

In-Vigo is based on a standard three-layer architecture virtualization (abstraction of resources):

First Layer: Creates a pool of resources namely virtual machines, data, applications and networks. Jobs are assigned to virtual resources, and virtual resources are controlled only through the domain.

Second Layer: Grid applications are instantiated as "services" that can be added if needed.

Third Layer: The value-added services are presented to users through portals and other channels. In this model, are used three portlets that are used for generating a dynamic interface, display and monitor the implementation of activities and to convey messages generated by the GAP.

It has three roles for users:

The first role is the single user who is responsible for interacting with the Grid portal using any of the three modules presented in the portlets. Users are clients (humans or other services) who use the legacy scientific applications. They have a working knowledge of how to use the application to be invoked, but not about the installation requirements of the application or about the underlying middleware involved in computing.

The second role is the expert user of the application. The user does not use the same user portlets above though using Grid-enabling instructions can access the application descriptor. This role is well informed about the implementation of the Grid -enable but does not know about the technical details of Web application middleware.

The other role of the application is the Administrator who is responsible for managing the Grid resources Grid and portals. They have an in-depth knowledge of Grid computing middleware and portal systems, even though users do not understand in-depth the performance the applications of Grid computing portals.

3.3 Toolkits

They are development tools that incorporate features to design portals grids.

3.3.1 Grid Portal Toolkit

It is a development tool for creating web portals and applications for Grid infrastructure [12], it was developed by Texas Advanced Computing Center (TAAC) [7], and its security model based on Globus GSI (Grid Security Infrastructure) for a single login and authentication to remote servers.

It allows to automatically incorporate the use of MyProxy, to be possible to authenticate users through identification and credentials for a short period of time [12], using the JSR168 standard.

3.3.2 Globus Toolkit

The Globus is a development toolkit used for building Grids. It was developed by Globus Alliance in 1998, the most popular versions incorporate GT 3.x OGSI (Open Grid Service Architecture), GT 3.9 and GT 4.0 supported by WSRF (WebService Resource Framework) [2].

Globus Toolkit includes features that can be used either independently or together with the application such as: security software, information infrastructure, resource and management, fault detection, among others.

4. Portlets and Servlet

Portlets are Web components based on Java or Java Server Pages that provide a modular structure that allows and facilitates the establishment of permissions to increase the granularity, and provide greater control over Grid portals [2].

Java Servlets are components of a web server with the role of providing impetus to the implementation, in this case to a Grid portal. According to [13], the difference between servlet and portlet is that portlets are multi-step and user-facing applications, portlets are used in Grid portals as pluggable interface components that form the user presentation layer, instead of the whole portal as Servlets do. [5].

The portlet will join the system stability based on low cohesion of services, in other words, whether the portlet that executes the functionality of a service is modified, deleted or a new one is added, other services are not totally shocked by this event.

The most widely used Java portlet standard JSP168 are called Java Portlet v1.0 and JSP286 called Java Portlet v2.0. One of the main objectives in the area of portlets is to maintain the interoperability of the data in the Grid portal. In [13], portlets are classified based on Data-based APIs.

Ownership of Data-based (based on data) allows portlets to share information among them in regard to how the concept of portlet application is introduced by the JSR-168 standard. This standard introduces portlets for Java platform designed interfaces, and the specific runtime behavior and relationship that should exist between a portlet and its container. The property-based API (based on APIs) can provide a programming interface for portlets, so that they can communicate their status to parties directly related to them. In addition, portlets can be used within the Grid as tools to apply ontology to Grid portal services [13]. This is accomplished by incorporating the standard languages OWL-S or BPEL4WS to the design of portlets so that it can capture data input, output and the process must form semantics.

Portals employ portlets as pluggable and related components that provide a presentation layer and user interface, allowing Grid portals to have a complete and easy to maintain structure by reducing the element of cohesion.

The standard JSR168 (Java Specification Request 168) is a portlet specification, in charge of maintaining interoperability between Grid portals and the portlets it uses. Thus its composition does not affect any of these or cause inconvenient in the implementation f a given service.

The JSR168 standard provides an API for portlets to generate URLs that exploit the context information provided during the encryption of the portal. JSR 186 Differentiates two types of URL Portlets: the navigation and resource [14].

Sometimes the portlet cannot run all the features needed in a Grid portal, in these cases, duality to the work is incorporated through Servlets [15].

The main advantages of the portlets include the following benefits: use the standard JSR-168, can be added or removed easily from a portal environment, can be chosen according to the needs of users. Portlets in a group of components can access multiple service groups [16].

Portlet and Servlet containers manage and e provide an environment for the execution of each of them.

Servlets communicate directly with their customers while portlets are invoked indirectly via the portal application. Portlets and Servlets are similar in many respects, with the main difference that Portlets can run in a portal environment, and servlets can only run on a default servlet container [17].

5. Discussion

The classification of the components of Grid Portals varies on the different papers presented; nonetheless, they coincide with the concepts of Frameworks, Middleware and Development Toolkits.

GridSphere Project as a framework is always considered as [2, 11, 18, 19, 20], but it is also considered a development toolkit [3]. Based on the characteristics stated Gridsphere is a framework for Grid portals because it is a prototype for the generation of certain web sites. It allows fast configuration of elements of the framework variants, such as: the core set of portlets that controls a wide range of portal services. In addition, it presents features for the development of other elements of Grid portals. Jetspeed 2 is a framework according to [2].

Other projects such as GridPort have been named differently by each of the quoted authors. Middleware is considered in [2], also [7] is considered a portal framework and in addition it is considered a development toolkit in [12]. Port is a based Grid Globus Grid Security Infrastructure that presents tools for the creation, development and configuration of portals; consequently, it can be considered a development toolkit.

Additionally, PortalLab is considered a Grid Portal [17] and a Toolkit in [3]. One of the features of PortalLab Development Toolkit is that it allows the creation of semantic portals consisting of Web Service oriented reusable portlets.

Globus is Toolkit [21] that has characteristics of Middleware and Development Toolkit. It is designed specifically for the development of grid portals including its components.

The In-Vigo project is a middleware [5, 10] based on its interconnection and independence characteristics. Astrolabe is a Grid Portal [4] that can be considered a Framework because it provides features similar to GridSphere as detailed in [4].

GTLab are libraries supporting workflows using a Directed Acyclic Graphs (DAG) Framework supported by the Globus Toolkit. Meanwhile, UDM Grid is a grid portal that does not correspond to a structure framework, or middleware toolkit, since it is a project designed for a specific use and presents the implementation of external elements to the same project.

6. Conclusion

Grid portals are intermediary applications between the user and the resources and services of the Grid. Portals consist of portlets, servlets and other Java components.

To design a Web portal, we use a portal framework, middleware, java components or development toolkits. These components are different in structure, architecture and features; hence, that the right and consistent name presented in the different works provides the best understanding of them. Portal frameworks are structures defined for a specific portal model, while middleware facilitates communication between structures, and toolkits allow the development of a portal from pre-developed tools.

References

[1] I. Foster, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," in Proceedings of the 1st International Symposium on Cluster Computing and the Grid (CCGRID ’01), 2001.

[2] Y. Cai, et al., "Portlet-based Portal Design for Grid Systems," in the Fifth International Conference on Grid and Cooperative Computing Workshops, 2006, p. 5.

[3] X. Chen, et al., "A Multilayer Portal Model for Information Grid," in The Third ChinnaGrid Conference, 2008, p. 8.

[4] H. Lin, et al., "Astrolabe: a Grid Operating Environment with Full-fledged Usability," The Sixth International Conference on Grid and Cooperative Computing (GCC 2007), p. 6, 2007.

[5] V. Sanjeepan, et al., "A Service-Oriented, Scalable Approach to Grid-Enabling of Legacy Scientific Applications," in IEEE International Conference on Web Services, 2005, p. 8.

[6] B. Bansal and S. Bawa, "Design and Development of Grid Portals," in IEEE Region 10th Conference, Melbourne, Australia, 2005, p. 5.

[7] J. Fang, et al., "Grid Portal System Based on GPIR," presented at the Second International Conference on Semantics, Knowledge, and Grid, IEEE, 2006.

[8] X. Meng, et al., "Applying Service Composition in Digital Museum Grid," Springer-Verlag Berlin Heidelberg 2005, 2005.

[9] X. Chen, et al., "Toolkits for Ontology Building and Semantic Annotation in UDMGrid," in the International Workshop on Metropolis/Enterprise Grid and Applications, 2005, pp. 486-495.

[10] S. Adabala, et al., "From virtualized resources to virtual computing grids: the In-VIGO system," Advanced Computing and Information Systems (ACIS) Laboratory, 2004.

[11] J. Novotny, et al., "GridSphere: An Advanced Portal Framework," in the 30th EUROMICRO Conference, IEEE, 2004.

[12] M. Dahan, et al., "Grid Portal Toolkit 3.0 (GridPort)," in Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, 2004, p. 2.

[13] O. Díaz, et al., "Improving Portlet Interoperability Through Deep Annotation," p. 10, 2005.

[14] O. Díaz and I. Paz, "Turning Web Applications into Portlets: Raising the Issues," in the 2005 Symposium on Applications and the Internet, 2005, p. 7.

[15] B. Beeson and A. Wright, "Developing Reusable Portals for Scripted Scientific Codes," in Proceedings of the First International Conference on e-Science and Grid Computing, 2005, p. 6.

[16] S. Trujillo, et al., "Feature Oriented Model Driven Development: A Case Study for Portlets," in 29th International Conference on Software Engineering, 2007, p. 10.

[17] M. Li, et al., "PortalLab: A Web Services Toolkit for Building Semantic Grid Portals," in the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003, p. 8.

[18] X. D. Wang, et al., "Flexible Grid Portlets to Access Multi Globus Toolkits," in The Fifth International Conference on Grid and Cooperative Computing Workshops, 2006, p. 6.

[19] R. Cao, et al., "USGPA: A User-centric and Secure Grid Portal Architecture for High-performance Computing," in International Symposium on Parallel and Distributed Processing with Applications, IEEE, 2009, p. 7.

[20] X. Yang and R. Allan, "Bringing AJAX to Grid Portals," IEEE, p. 8, 2007.

[21] F. Ionescu, et al., "Portlets-Based Portal for an E-Learning Grid," CIMCA 2008, IAWTIC 2008, and ISE 2008, 2008.

Xavier Medianero-Pasco is a student of the Master of Science in Information and Communication Technology at the Technological University of Panama. He was awarded a Bachelor’s degree in Engineering and Computer Science from the Technological University of Panama in 2009. His research interests include Aspect Oriented Programming Framework, Grid Computing and other topics.

Belen Bonilla-Morales is a student of the Master of Science in Information and Communication Technology at the Technological University of Panama. She was awarded a Bachelor’s degree in Engineering and Computer Science from the Technological University of Panama in 2009. Her research interests include Software Engineering, Semantic Web and Grid Computing.

Miguel Vargas-Lombardo is a professor at the Technological University of Panama, PhD awarded by the Polytechnic University of Madrid. His main research lines are gird computing, health-grid and e-health.