Uni-Grid P&T: A Toolkit for Building Customizable Grid Portals*

Jieyue He[1], Hongqiang Rong[2], Zongwei Luo[2], Joshua Zhexue Huang[2], and Frank Tong[2]

[1] Department of Computer Science, Southeast University
Nanjing, China
jieyuehe@seu.edu.cn
[2] E-Business Technology Institute
The University of Hong Kong, China
{hrong,zwluo,jhuang,ftong}@eti.hku.hk

Abstract. This paper presents an architecture and functional design of Uni-Grid P&T, a toolkit for building customizable Grid portals for different application domains in a typical university environment in China. Based on a layered architecture, the functions of the toolkit are designed on top of the new Open Grid Services Architecture (OGSA) and based on the standards of Web services. We particularly address the issues of collaborative portals and portlet implementations of the toolkit. The development of the toolkit is part of the initiative of the ChinaGrid project launched in 2002 by the Ministry of Education of China that is aimed to build a China national education Grid to link more than 100 major universities across China.

Keywords: Grid computing, Grid portal, Grid portal toolkit, Web services.

1 Introduction

After a decade of development and experiment, Grid Computing has gained a wide acceptance as a new infrastructure for computing and information management [1,2]. The Grid is an innovative model of distributed computing focusing on large-scale resource sharing for data processing, collaborative applications and high performance computing[3]. The resource-sharing model promises researchers and scientists in all disciplines to access and use HPC (high performance computing) facilities to solve their research and engineering problems.

In China, the HPC development is facing a dilemma[4]. On one hand, many researchers and engineers need HPC facilities but do not have the access due to the centralized control of these facilities and the lack of specific skills to program and operate. On the other hand, the computing facilities are often not

* The work was conducted when the author was visiting the E-Business Technology Institute of The University of Hong Kong, under the support of the IBM China Scholar Visitorship Program and the China's National '863' project (#2002AA231071).

M. Jeckle and L.-J. Zhang (Eds.): ICWS-Europe 2003, LNCS 2853, pp. 198–212, 2003.

fully utilized because of lack of users and job submissions. The valuable resources are idle and wasted. We see the Grid sharing model offers to fill the gap between the great demand from the large user community and the scarcely accessible high performance computing resources.

In this paper, we propose a platform and tools that can be used to build Grid portals for different disciplines in a typical university environment in China. We name it as Uni-Grid P&T. Building on top of the OGSA infrastructure[11][25], the Uni-Grid P&T provides an open architecture and a set of tools which allow rapid development of customizable portals in different application domains, such as physics, chemistry, biology and bio-informatics, etc.

The Uni-Grid P&T initiative is a part of the ChinaGrid project that was launched by The Ministry of Education (MoE) of China in 2002. The first phase of ChinaGrid is to connect 12 universities through the MoE's broadband network (CERNET). The final goal of ChinaGrid will connect 100 major universities across China and link ChinaGrid to Hong Kong. The project itself is a great challenge because of the vast geographical region and the diversity of various resources. As part of it, the Uni-Grid P&T will be implemented at the Southeast University (SEU), China. SEU is participating in the first phase of ChinaGrid project. Among the 45 departments and research centers in SEU, there are good many setups of high power servers and HPC systems. These facilities will be connected in the campus Grid which is planned to become a node in ChinaGrid.

To give a broader access to university students and researchers in different disciplines, the key solution is to build different Grid portals for them to use. To close the gap between the complicated Grid environments and the common users, provision of portals with easy-to-use interface and user familiar services is essential. Although there are toolkits available for building Grid portals[5], e.g. Cactus[6], Gateway[7], Unicore[8], Legion[9] and Gridlab[10], they are not entirely suitable to the diversity of users and application domains in China in the considerations that (i) most portals are domain-specific, and cannot be shared efficiently, (ii) cross-domain portals normally require high budget and large supporting staff, (iii) many projects have become outdated in face of the rapid development of Grid technologies, for instance the OGSA[11][25] and Web services[12], and (iv) most issues of deployment and administration in portals are not well addressed[13].

Our goal is to develop a campus Grid platform through which university students and staff can access the Grid from their own terminals and stations. We will provide them the users tools for rapid development of customized portals for different disciplines. New portals and services can be readily built on top of existing services offered in the campus Grid. Serving portal needs of different disciplines, our work will reference the techniques of user specific Grid portal[14] but adapt with focus on the provision of domain-specific portal functionality. In particular, our proposed platform and tools will fulfill to address the following issues:

(1) Rapid development of domain specific portals for university students and researchers. These portals will provide the basic Grid services, and will include

application specific services that are needed in solving domain problems in various disciplines such as in bio-informatics, GIS, physics, etc.

(2) Rapid development of new applications on top of the basic Grid services. We will develop a graphic tool that allows users to build high-level applications by integrating a process of the basic Grid services. For example, a data-clustering algorithm can be implemented as a basic Grid service, and domain specific portals can provide services accessing it.

(3) Portal aggregation that creates new portals by aggregating services from multiple portals to support cross-disciplinary applications.

(4) Client environment that is used to develop basic Grid services such as coding a parallel clustering algorithm, compiling and registering it with the service resources.

This paper is organized as follows: In Section 2, we present the major features and services to be provided in the Uni-Grid P&T. In Section 3, we describe the Uni-Grid P&T architecture. In Section 4, we discuss the technical issues. We illustrate on the topic of collaborative portal customization in Section 5. The related projects are discussed in Section 6. The conclusions and our future work are given in Section 7.

2 Portal Services

2.1 Services

We classify services accessed from the Grid portals into three categories: *basic generic services, basic domain specific services and super services.*The basic generic services are those services that provide common Grid operations. Examples are submissions service and data service. The basic domain specific services are those services that provide basic operations required in a particular domain. For example, geographical data conversion from vector presentation to raster presentation is a basic domain specific service in GIS. The super services are those services that are built of a set of basic services to provide high-level services. The super services can be either generic or domain-specific. Domain application developers use a graphical interface and a visual definition language to create domain specific super services.

The basic generic services are divided into the following three groups: *core services, development management and administration management.* Major services in each group include the following.

(1) Core Service
Security Service – Users logon to the portal with a secure Web form to gain access to services provided by the Grid. Each site can restrict and limit accessible resources at each target system, thus retaining the ultimate control on the resources.

Submission Service – The users can specify the host on which they run their jobs. For parallel jobs, we provide a graphic interface for the user to interactively create the execution process of mutually linked jobs and submit the process to the Grid. A task is divided into subtasks that form an execution tree. In the graphic interface the user can use a mouse to select different services for different subtasks and link them into a process. The Resource service first selects the most appropriate resource for a given job taking into account factors such as cost, CPU speed and availability. Then the File service is used to send jobs to different computers for execution.

File Service – This service is used to transfer the files in different resources. Monitoring service – To track and monitor the job execution, the job status function timely displays the job status with different colors of the subtask icons on the interface. The job states include waiting, executing, finished and failed. All the states are saved in the system logs for error checking.

Data Service – Users on the Grid not only share computing resources but also data resources. These resources are usually located in different geographical locations. The geographical locations of the resources are transparent to users. The data about the resources themselves and their usage include data about job executions needed and data created by the job upon its finish. Therefore, at job submission, the data service is used to select data for the job and specify the location to store the result.

Information Service – Because the user can only select the machines authorized to run his/her jobs, the resource availability status must be provided. This information includes the status of batch queuing system, load, and network performance between the resources.

Workflow Service – This service is used to define a procedure (workflow) comprised of one or multiple services. Thus, services in different domains could complete a task coordinately.

(2) Development Management

Application Service Customization – When a domain wants to share an application as a service, because of domain diversity, users can customize the service for their domains by application service customization.

Services Generation – Web services are designed to provide application integration via the use of a standard mechanism to be described, published, discovered, invoked, and composed automatically. Reusability and cooperation are trends of science technology development. A function should be converted into a Web service when it is shared in more domains. Services generation is an ability to convert functions of different domains into executable Web services so that they are accessed or called by other users with different languages.

Factory Service – Factory service is used to produce a service instance for a portlet. Portlet will be explained in section 5.

(3) Administration Management
Account Service – Manage users, including adding, deleting and modifying users.

Resource Service – Manage computers, large volume storage devices, data resource, etc.

2.2 Role Based Access to Services

The services can be used in portlets configured differently and aggregated into different portals for different application domains for different usages. We divide users who access to and use the Grid resources into four categories, each with a different role in the Grid service development and usage.

Grid Administrators manage users and resources including computing resources, data resources and service resources. Administrators are also responsible for maintenance, update and expansion of these resources, such as adding a new computing node to the Grid. The additions of new resources are reflected in the resource directory so the users can find and use them.

Service Developers include both basic generic service developers and basic domain specific service developers. As applications increase, more services are required. The basic generic service developers develop and add new basic services to the service directory for application developers to choose in implementing their applications. Frequently application developers need to convert non-Grid applications to Grid services. For example, when a computational biologist wants to share his algorithm code with other biologists on the Grid, he can make use of service generation to export it as a Web service and add it to the resource Grid service directory so other users can use this service.

Application Developers use the Grid services provided in the Grid portal to develop new applications using a graphic interface. A visual definition workflow language is used to connect a sequence of basic services to perform a computation operation. If the new application can be reused, it can be added to the portal as a super service. The requirements for the graphic interface are:

- A user friendly environment that can be accessed anywhere, anytime using different terminals such as PCs, PDAs and other mobile devices.
- It is critical to hide the complexity of both environment and applications from the operators. Application users should be free from (1)knowing

the geographical locations of resources, (2) frequent separately logon and security checking of every resource used, (3)manually submitting jobs to particular computer systems and (4)keeping track of the status of computational tasks operating on the resources, or with reacting to failure.

Information Viewers only use portals to search for execution results of some jobs. For example, stock researchers who study a data model and develop programs construct the stock prediction application before it is submitted to a Grid to run. The execution result changes over time. Once it is configured and submitted, the Grid service can be provided to the users who use Internet browsers to view the results.

3 Architecture

Nowadays, Web services have become increasingly important in the enterprise community and many new standards and implementations are emerging. The Web Services Definition Language (WSDL)[16] permits services to be defined in a standard interface language and registered and discovered using the Universal Description, Discovery and Integration (UDDI)[17] technologies. The Simple Object Access Protocol (SOAP)[18] provides a standard for communicating structured information using XML. Building on such open foundations Web services have become an appealing technology for system and application integration. The OGSA, a recent trend for enabling the service oriented Grid has adopted web services technologies as well. In our Grid portal work, we also find it a right approach to building a Grid portal customization environment based on the service oriented architecture with a discovery and binding mechanism that can be used to deal with customizing and adding new functional services. This architecture provides abilities to connect several functional services that perform the tasks demanded by the application.

Fig. 1 illustrates the Uni-Grid P&T architecture. It is a layered structure. The lowest level is the Grid fabric resources distributed in the virtual organizations. The second lowest level is the Globus toolkit 3.0[19]compatible with OGSA. The Globus toolkit is the most widely used Grid middleware systems. It provides a small set of useful services, including authentication, remote access to resources, and information services to discover and query such remote resource. Unfortunately these services may not be compatible with the commodity technologies used for application development by software engineers and scientists, and do not provide much direct support for building Grid Computing Environment (GCE). Above Globus Toolkit is the Java Commodity Grid Kit (CoG)[20], which provides a Java Commodity Grid interface to the Gloubs toolkit. On the top of COG are the various services provided to users.

In order to provide the customization capabilities, services offered to end users will be provided in the form of portlets[21] contained in the portal. Portlets have become an increasingly popular concept used to describe visual user interfaces to a content or service provider. From a user's perspective, a portlet is

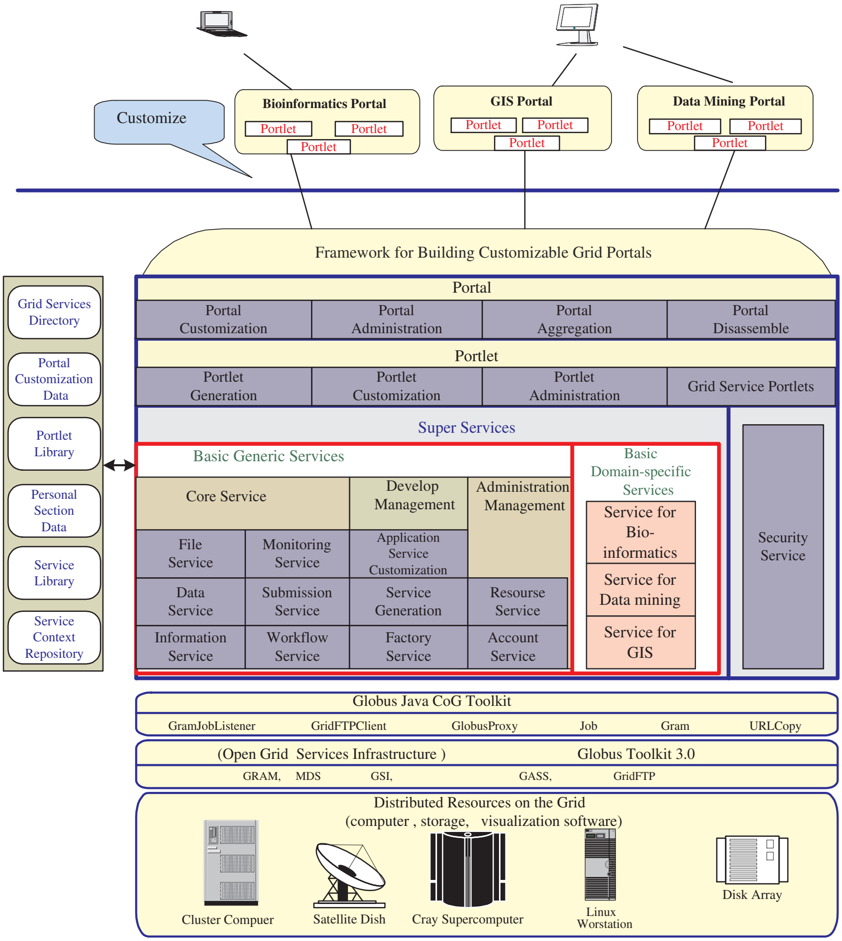

Fig. 1. Architecture of the Uni-Grid P&T.

a small window in the portal page that provides a specific service. From an application development perspective, portlets are pluggable modules that are designed to run inside the portlet container. Consequently, the portlet tier is on top of the services layer. Because all the Grid services will be used as portlet, the services will be converted into Grid service portlets by portlet generation. The portlet can be customized through the portlet customization process. The functions of portlet administration are used to publish, query portlets by UDDI, add and delete portlets.

Many portlets can be aggregated in a portal, so on the top of a portlet layer is the portal tier. It includes portal customization, portal administration, portal aggregation, and portal disassembling. Portal customization is used to customize a domain portal for a specific application. When the customization finishes, the portal configuration parameters will be saved in customization portal data. The portal administration service is used to select the portlets from the Grid services directory, search and delete portlets. Portal aggregation is used to aggregate portlets that have been selected into a portal. The portal as well as portlet information is published in the Grid services directory. When a user is customizing his/her portal, he may not only select the portlets, but also select the portal that has been aggregated. In this case, the original portal will be disassembled into portlets. Then these portlets will be aggregated to form a new portal.

The Uni-Grid P&T contains two types of portals. One is the portal formed when The Uni-Grid P&T is realized, i.e., the portal customization portal. The other type includes those portals customized through the customization portal.

Besides the usual mandatory metadata for describing resources, the Uni-Grid P&T also requires portal customization data, which is about the configuration parameters of a portal. The personal session data are also saved for each user to restore his last normal status when he logs in again to the portal. Since the Uni-Grid P&T is extendible, more services can be added and aggregated dynamically. These services are stored in the services library. The library itself can be distributed in the Grid nodes. The services are provided in the form of portlets to the service users. Each service has a corresponding portlet that is stored in the portlet library. Logically the portlet services will be published in the Grid service directory. Every service has a service context, which describes how this service is to be invoked. All the context information is saved in the service context repository. For example, when a new data mining cluster algorithm is encapsulated in a service, scientists must know the data format, parameters setting, and other configuration in order to use the algorithm to predict protein secondary structures.

The architecture is flexible. It has following advantages: (1)The functions provided by domains experts could be distributed geographically. Thus the authors of functions are able to maintain and deploy them with little cost.(2)Users can extend their services to the Uni-Grid P&T as necessary. (3)Users can easily replace a service, making it effective to expose a customized functionality to disparate user communities.

4 Technical Issues

As the Uni-Grid P&T is open, new functions can be easily converted into a service by service generation and added to the open system. Thereon, the Uni-Grid P&T will generate the portlets automatically for good efficiency and user convenience for the portlet creation. At the same time, the service context repository and the resource description are also inserted automatically. The process is illustrated in Fig. 2.

In Fig. 2, the service description gives an account of the service metadata and the interface information defined and published by the service generation. The services context repository tells about how to use the service such as the type of input data, the steps and so on. The resource description indicates where the service is located physically.

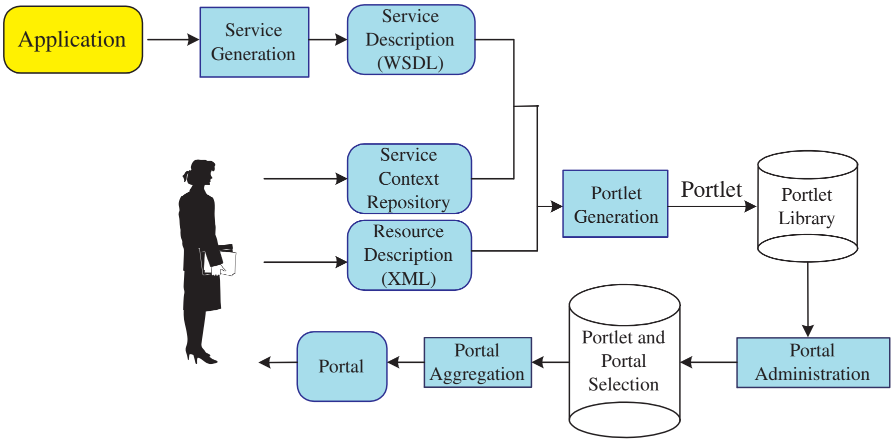

Fig. 2. The relations among an application, services, portlets and a portal.

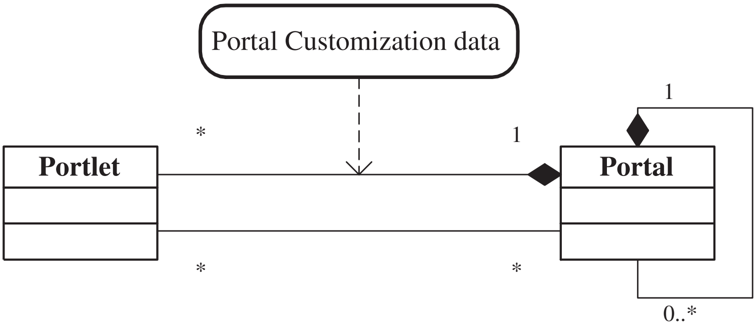

Fig. 3. The relation between a portal and a portlet.

In our framework, one can construct a portal by aggregating the existing portals as illustrated in Fig. 3. The process usually involves portal disassembling and aggregation. The underlining mechanism for portal aggregation is implemented through the communication scheme among portlets. For example, when a job is submitted through the submission portlet, the job id will be sent to the monitor portlet, so that the running status of the job can be shown and tracked. The process is conducted through the mechanism of the portlet event publication and consumption. As illustrated in Fig. 4, a portlet event coordinator is triggered when the portlet message/event arrives. The message/event listener detects the occurrence of the message, and the portlet handler provides the interface to handle the portlet event. For example, when the user generates a portlet event (he/she clicks a submission button), the portlet event message will be sent to the portlet event coordinator and then forwarded to the portlet handler.

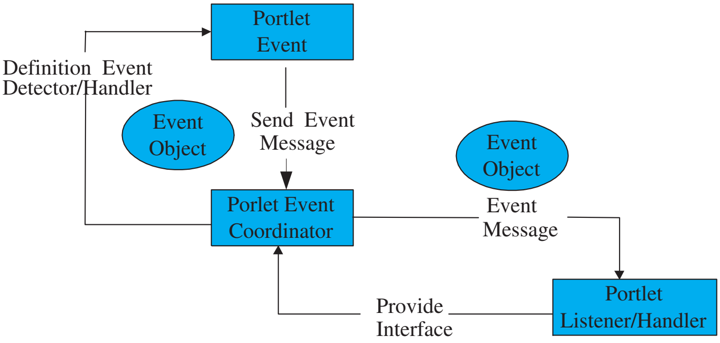

Fig. 4. The communication among portlets.

The current implementation of Uni-Grid P&T is based on the Apache projects Tomcat, Jetspeed and Axis[26]. Tomcat provides Web Application Server environment and a J2EE Web container. Jetspeed provides the flexible and reconfigurable portal environment. Axis running in a J2EE Web container provides a SOAP message listener (Axis Servlet) and is responsible for SOAP request/response serialization and de-serialization. WSDL2Java converter is used to generate the client side and server side code for the target Web service. There exist many services in the platform, each having a language associated with it. For example, in the security service, WS-Security protocol stack is used to describe the security policies and in the workflow service BPEL4WS[24] is chosen and remote portlets are enabled by Web Services for Remote Portals (WSRP)[23].

5 Collaborative Portal Customization

The Uni-Grid P&T can be used not only for customizing a portal for a specific domain, but also for a super collaborative portal across many application domaions. In this section we describe how to use the Uni-Grid P&T to make collaborative portal customization. We illustrate the steps using an example in bioinformatics application domain. The customization process can be generalized to other domains as well. Bioinformatics is a multi-discplinary field requiring collaboration efforts. Reserach activities like Gene Regulatory research, DNA sequence assembly, prediction of protein secondary structures invovles mathemtical, statistical and computer systems to analyze biological data. Bioinformatic applications use computer systems to the store and analyze genomic and proteomic sequences and structures. All of them require huge computing power and mass storage in which the Grid technology is an ideal candidate. At the same time, it is necessary to have a portal to facilitate such collabtiavtive activities. Such a portal usually requires customizations for the target usage.

For example, when an application developer wants to build a Grid portal for the users in bioinformatics, he first needs to logon to the customiation portal. Once he is authenticated, all the actions authorized for him will be made available. He will be permited to check the customization portal data to see if the desired portal has been built. Each service is encapsulated as a portlet and published in a Grid services directory. Thus, if the developer does not find the desired portal, he can query the Grid services directory and find out the ones he needs. The developer selects necessary portlets through portal administration service and then can customize them. The portal configration data will be added into the portal customization data as part of the user portal profile. The process is showed as the left part of Fig. 5.

Bioinformatics research requires the collaboration of experts in multiple domains. Uni-Grid P&T provides a collaborative protal to realize such mechanisms. For example, in the gene regulatory research, it is necessary to establish the mathematical evaluation model for the relativity of the gene representation models, through the aggregation analysis of the gene representation data. The procedure includes the Gene regulatory information retrieval, data cleaning, cluster analysis, relevant mathematic model building. If a new cluster analysis algorithm is provided, it must be converted into a service through the service generator and saved into the service library. At the same time, it will be converted into a portlet for other users to use. Because the functions provided by domains experts could be distributed geographically, obviously, it would be much more convenient if remote Web services would appear as remote portlets including presentation and application logic[22] by WSRP[23]. These portlets will be aggregated to a portal that can be invoked through a standard interface using generic portlet proxies on the portal side. This means that no special portlet code needs to be installed on the portal. Use of generic portlet proxies eliminates the need to develop specific portlets for each service to run on the portal. Through the use of portlet proxies, the remote portlets appearing on the portals just like local portlets and can be selected by users easily. The services provided by domains experts can be distributed geographically and published as remote porlet Grid services in the Grid services directory as shown in Fig. 5. For example, data mining experts can publish a clustering function as portlet in the Grid services directory. Once the remote portlet is published, the bioinformatics researcher can use the portal administration service to search the Grid services directory, select the portlets needed, add them into the portlet and portal selection library, and then generate a portlet proxy used for calling remote porlet Grid services. Therefore, for a specific application domain portal, it includes two portlets: local portlets and remote portlets, which run on the local portal server and the Grid as remote services respectively.

The Gene Regulatory research constitutes many steps, connected in a process flow that is supported with the workflow service provided by the customization portal. The flow can be described in Business Process Execution Language for Web services (BPEL4WS[24]). The BPEL4WS describes a flow as a sequence of activities performed by various service providers (partners) and tied together through sequential control links. The activities, like other services, can expose their interfaces described using the standard WSDL for external use. On execution, the sequential control links will trigger the correspondent portlets and the user can use these portlets to perform his computational tasks.

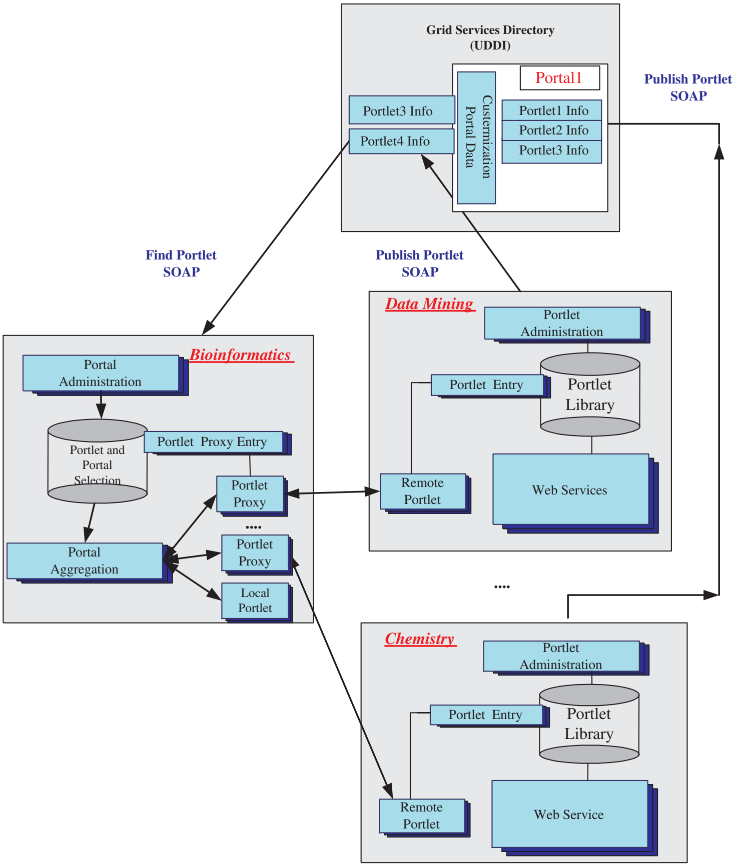

Fig. 5. Collaboration and remote portlet Web services.

6 Related Work

Development of suitable toolkits of higher-level Grid services is considered as an important catalyst to boost the use of the Grid technology and facilities[10]. In this section, we review three well-known pioneer toolkits that provide excellent references to our own work.

GridPort[15], a Grid portal toolkit for building Grid computing portal, was developed from the HotPage project carried out at the San Diego Supercomputer Center (SDSC) in late 90's. GridPort was generated from a subset of software that provides account management and resource usage functions for the HotPage portal. GridPort can support multiple application portals on a single Grid with a single login environment. Because it is simple, robust, and flexible, GridPort has been used to build several science portals such as the Laboratory for Applied Pharmacokinetics modeling portal at the University of Southern California, The Telescience portal at the National Center for Microscopy and Imaging Research at UCSD and the NBCR computational portal at SDSC. GridPort is one of the early portal toolkits implemented in Perl language. Its early design was not based on the new Open Grid Services Architecture (OSGA) and Web services standards.

The Grid Portal Development Kit (GPDK) was developed at Lawrence Berkeley National Laboratories (LBNL)[14]. GPDK is aimed to provide generic user and application portal capabilities and was designed with the following criteria: the generic, reusable and common components access the Grid services supported by the Globus middleware; it uses customizable user profiles to support easy creation of application portal specific profiles; it provides a complete development environment for building customized application specific portals based on the core set of the GPDK Grid service components; it provides an open architecture to leverage open source software technologies. GPDK provides a suite of JavaBeans suitable for the Java based GCE environment and was designed to support the JSP (Java Server Pages) display. The development of GPDK is moving to adopt the new standards of Web services.

The Grid Application Toolkit (GAT) is an on-going effort that is aimed to provide an application programmer with a single interface to the Grid environment and allows the programmer to easily develop applications. GAT is the main deliverable of the GridLab project, a joint European Grid effort[10]. GAT will provide functionality that enables an application to call the underlying Grid services through a carefully constructed set of generic high-level APIs. Unlike GridPort and GPDK, GAT is more focused on APIs to the underlying Grid services, rather than functions of building application portals. Therefore, GAT is a tool for service developers to implement domain specific basic Grid services.

Compared with the above related work, we are focused on collaborative portals to enable multi-disciplinary applications and inter-disciplinary collaborations. In this Uni-Grid portal framework, portal toolkits are provided to realize the wrapping of applications into web services. Then those services are aggregated in the form of portlets into a collaborative portal according to application requirements. Furthermore, individual collaborative portals can be aggregated into a new collaborative portal.

7 Conclusions and Future Work

The Grid is rapidly evolving in concepts, technologies and implementations. However, the ultimate success of this innovative computing model and infrastructure highly depends on its attraction of users in large numbers to use it and benefit from it. The Grid portal is moving the way to offer easy access to the Grid services. Development of the toolkits for building various portals in an economic way and offering opportunities to application developers to develop their own domain specific services is becoming a focal point in the Grid technology development. The Uni-Grid P&T design is our first step to develop a suitable tool for use of university research students and staff in developing Grid technologies and applications. We have built a test Grid environment consisting of an IBM

SP2 system, IBM pSeries 660 and 670 high-end servers, and a cluster of Linux PCs communicated with the Globus middleware. Since the framework is very huge, there are many services to be implemented. To accomplish all these purposes, we plan to partner with other ChinaGrid participating universities. The core services will be implemented first along with other development management services, like service generation and administrative management services to construct the basic Grid computing platform. More and more domain specific services including super services will be added accordingly when this platform is used to develop applications to enable more and more inter-disciplinary collaborations.

References

1. David De Roure, Mark A. Baker, Nicolas R. Jennings and Nigel R. Shadbolt, "The Evolution of the Grid", in Grid Computing – Making the Global Infrastructure a Reality, ed. by Fran Berman, Geoffery Fox and Tony Hey, John Wiley & Sons, 2003, pp. 65–100.
2. Fran Berman, Geoffrey Fox and Tony Hey "The Grid: past, present, future", in: Grid Computing – Making the Global Infrastructure a Reality, ed. by Fran Berman, Geoffery Fox and Tony Hey, John Wiley & Sons, 2003, pp. 9–50.
3. Ian Foster, C. Kesselman and S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organization", Int. J. of Supercomputing Application, 15(3), 2001.
4. Liu Peng, Li San-li, Huang Zhen-chun, Du Zhi-hui, He Chuan and Chen Yu, "xGrid: a Grid Prototype for Interdisciplinary Research", To appear in Minicomputer and Microcomputer System, 2003 (in Chinese).
5. Geoffrey Fox, Dennis Gannon and Mary Thomas "Overview of Grid Computing Environments", in: Grid Computing – Making the Global Infrastructure a Reality, ed. by Fran Berman, Geoffery Fox and Tony Hey, John Wiley & Sons, 2003. pp. 544–553
6. Cactus http://www.cactuscode.org/
7. Marlon E. Pierce, Chonhan Young, and Geoffrey C. Fox, "The Gateway Computational Web Portal", Concurrency and Computation: Practice and Experience Vol. 14, No. 13–15 (Grid Computing environments Special Issue), pp. 1411–1426, 2002.
8. Dietmar W. Erwin, "UNICORE-A Grid Computing Environment", Concurrency and Computation: Practice and Experience Vol. 14, No. 13-15 (Grid Computing environments Special Issue), pp. 1395–1410, 2002.
9. Anand Natrajan, Anh Nguyen-Tuong, Marty A. Humphrey, Michael Herrick, Brian P. Clarke and Andrew S. Grimshaw, "The Legion Grid Portal", Concurrency and Computation: Practice and Experience Vol. 14, No. 13–15 (Grid Computing environments Special Issue), pp. 1365–1394, 2002.
10. Gabrielle Allen, Kelly Davis, Konstantinos N. Dolkas, Nikolaos D. Doulamis, Tom Goodale, Thilo Kielmann, Andrè Merzky, Jarek Nabrzyski, Juliusz Pukacki, Thomas Radke, Michael Russell, Ed Seidel, John Shalf and Ian Taylor. "Enabling Applications on the Grid: A GridLab Overview", International Journal of High Performance Computing Applications: Special issue on Grid Computing: Infrastructure and Applications, to be published in August 2003.
11. Ian Foster, Carl Kesselman, Jeffrey M. Nick and Steven Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." June 22, 2002. http://www.globus.org/research/papers.html

12. Web services http://www.w3.org/2002/ws/
13. Gregor von Laszewski, Eric Blau, Michael Bletzinger, Jarek Gawor, Peter Lane, Stuart Martin and Michael Russell, "Software, Component, and Service Deployment in Computational Grids", The First International IFIP/ACM Working Conference on Component Deployment, June 20–21, 2002 Berlin, Germany
14. Jason Novotny "The Grid portal development kit", in: Grid Computing - Making the Global Infrastructure a Reality, ed. by Fran Berman, Geoffery Fox and Tony Hey, John Wiley & Sons, 2003, pp. 657–673.
15. Mary P.Thomas and John R. Boisseau "Building Grid computing portals: NPACI Grid portal toolkit", in Grid Computing - Making the Global Infrastructure a Reality, ed. by Fran Berman, Geoffery Fox and Tony Hey, John Wiley & Sons, 2003, pp. 675–699
16. Web Services Definition Language http://www.w3.org/TR/wsdl
17. Universal Description, Discovery and Integration http://www.uddi.org
18. Simple Object Access Protocol. http://www.w3.org/TR/SOAP/
19. The Globus Grid Project http://www.globus.org
20. The Java Commodity Grid (Java CoG) Kit, http://www.globus.org/research/development-environments.html
21. What is a Portlet? http://www-3.ibm.com/software/webservers/portal/portlet.html
22. Thomas Schaeck "WebSphere Portal Server and Web Services White paper", http://www-3.ibm.com/software/solutions/webservices/pdf/WPS.pdf/
23. Web Services for Remote Portals http://www.oasis-open.org/committees/wsrp/
24. BPEL4WS Business Process Execution Language for Web Services http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/
25. Ian Foster and D. Gannon, "The Open Grid Services Architecture Platform", Feb 16, 2003, http://www.ggf.org/ogsa-wg
26. http://www.apache.org/