

Integrating Grid and Web Services: A Critical Assessment of Methods and Implications to Resource Discovery

Aisha Naseer
School of Information Systems,
Computing and Mathematics
Brunel University, Uxbridge UB8
3PH, United Kingdom
+44 (0)1895 274000
aisha.naseer@brunel.ac.uk

Lampros K. Stergioulas
School of Information Systems,
Computing and Mathematics
Brunel University, Uxbridge UB8
3PH, United Kingdom
+44 (0)1895 274000
lampros.stergioulas@brunel.ac.uk

ABSTRACT

Web technologies have played a significant role in supporting global sharing of Internet resources and thereby improving communications. Grids hold the promise to provide global interoperability and interconnectivity at a level considered impossible a few decades ago. In practice, there is not much difference between the existing Grid and Web infrastructures; in fact, a Grid infrastructure could be built by making few modifications to a Web infrastructure. The implementation of Web-based Grids or a partially-Gridified Web is one of the potential solutions to Grid infrastructure problems. This can be done by sharing Grid services across the Grid infrastructure using the underlying Web services as vehicles or transporters of these services. In this study we argue that Grids should be developed using the underlying Web infrastructure and Grid services could be integrated with Web services using inheritance techniques to produce Grid-supported Web services. Furthermore, this approach seems to deal effectively with the problems of resource discovery in such partially-Gridified Web environments.

Categories and Subject Descriptors

H.3.4 [Web]: World Wide Web or the Internet

I.7.2 [XML]: Extensible Markup Language

General Terms

Management, Performance, Reliability, Security, Standardization.

Keywords

Grid Services, Web Services, Resource Discovery, Encapsulation, Inheritance, Gridified Web.

1. INTRODUCTION

Web technologies have played a significant role in supporting global sharing of internet resources and thereby improving communications. Grids are expected to power up the next generation of Web and hold the promise to provide global interconnectivity and interoperability regardless of the geographical and heterogeneity constraints and at such an advanced and ubiquitous level that was considered impossible a few decades ago. Grids enable the sharing, selection and aggregation of geographically distributed 'autonomous' resources dynamically at run time depending on their availability, capability, performance, cost, policies or rules which govern them and the users' quality-of-service requirements and reliability. Grids are composed of a global network infrastructure that

contains several other networks (local or wide area) and exhibits extensively high synergy among all its units (humans, networks, programs, processes and services, etc). Cross-communication among the various types of units is needed on a large scale in order to enhance the capabilities and capacities of Grids. Such units or resources are capable of sending and receiving requests/commands and act as independent communicators. This can be made possible by specialized services that are available on the Grids (Grid services).

The trend to use Web-services as wrappers or transporters of Grid services is today becoming more and more popular. Since Grids can be viewed as the next generation of Web, Grid infrastructure could be built by making minor modifications to the Web infrastructure. The implementation of Web-based Grids or partially-Gridified Web is one of the potential solutions to Grid infrastructure problems, which would facilitate the formation of virtual organizations globally. These virtual organizations would be Grid-like structures over the Web.

Grid services provide a means for discovering other Grid resources; the term used for this operation is Resource Discovery. The Grid community has not yet been able to fully resolve the issue of Resource Discovery due to various technical constraints and geographical limitations. Grid services can be shared across the Grid infrastructure by using the underlying Web services as vehicles or transporters. Integration of Grid services into Web services can provide an efficient way to pass requests and responses to resource discovery queries, which is a key issue to be addressed. There are two potential ways to address this issue: (a) by using Encapsulation techniques and (b) by using Inheritance techniques for services integration. The former encapsulates or wraps Grid services into Web services as specialized functions and the latter inherits or extends Grid services from Web services to support service integration.

There is a well recognized need for both the Grid and Web communities to work in close collaboration and find out a Web-based solution to Grid resource discovery problems. This paper discusses the main issues and examines potential solutions. Section 1 offers an introduction to the problem, the second section discusses Grids as the next generation of Web. Section 3 defines Grids services, Web services and their integration. Grid resource discovery issues and methods are reviewed in Section 4. Section 5 argues that a potential solution to Grid resource discovery lies in the integration of Grid services with Web services by using the Grid-based Web service approach or the Web-based Grid service approach, depending on the respective viewpoint. A review of related work towards achieving this goal is included, and the two

approaches are examined and suggestions are made as to the suitability of these approaches if implemented in their respective environments by employing encapsulation or inheritance. Conclusions are drawn in the Section 6.

2. GRIDS - THE NEXT GENERATION OF THE WEB

Grids are Multi-Peer to Multi-Peer network architectures [1] on which all the components (resources, protocols, interfaces, services, network layers, etc.) are integrated with each other in a controlled and consistent manner so as to support the functionality of the Grid. The primary purpose of Grids is to globally share resources across various organizational domains regardless of the geographical dispersion. The Grid infrastructure, being distributed in nature, allows for high variability in user and resource participation. It deploys a decentralized computing environment, which is compatible or interoperable with all sorts of network architectures. Grids aim to provide abstraction - at both the user and resource level - that is transparent to the user and relies on a standard-based service infrastructure to share computers, storage space, sensors, software applications and data, etc. across organizational boundaries. The Web can be considered as an "information Grid" and the Grid an "extended Web" that goes beyond information sharing to allow users to share computer resources [9]. In fact, the Web infrastructure is quite similar to the Grid infrastructure in a way that it aims at interconnecting hardware, software and users for sharing via some transmission media and provides abstraction and transparency at the user level with limited functionality or accessibility options. Whereas, in addition to the services provided by the Web, Grids are expected to possess extra capabilities and features, such as:

- a. Automatic online upgrading of resources at runtime - (Self-Management)
- b. Automatic online configuration of settings - (Self-Configuration)
- c. Automatic online scheduling - (Self-Motivation)
- d. Automatic dynamic expansion (addition of new nodes) - (Self-Developed Growth)
- e. Automatic online repairing and elimination of distortion among resources or nodes - (Self-maintenance)
- f. Intelligent synthesis of knowledge and creation of security models for Grids dynamically
- g. Intelligent security system that dynamically generates automatic alerts throughout the Grid
- h. Automatic tracking of resources - (Resource Discovery)
- i. Automatic intelligent distributed resource management - (Self-Integration)
- j. Automatic resolution of semantic compatibility issues

Moreover, the types of resources available on a Grid are generally more powerful, more diverse, and better connected than the typical Web resource. The service type resources over the Grids are generally more complex, since they have to support dynamic environments and communities. The maturity state of Web

framework convinces implementation of Grid infrastructure over the Web infrastructure instead of implementing a totally new Grid infrastructure that has a number of issues. By providing specialized services onto the Web and making them extensible with additional features such as the global provision of computational power, storage space and inter-group collaboration across various geographically dispersed domains would enable the building of Grid infrastructure over existing Web infrastructure, thereby making it a "partially Gridified Web" (see Figure 1). It would possess all the features and functionalities of a Web with some additional and enhanced functionalities to make it "Grid-enabled".

A Web-based Grid infrastructure would transform the basic elements, such as Extensible Markup Language (XML), Web Service Definition Language (WSDL), Universal Description, Discovery, and Integration (UDDI), and Simple Object Access Protocol (SOAP) Web services, etc., into extensible carriers of specialized (Grid) services that would be dedicated to dealing with requests for specialized tasks on the Grids.

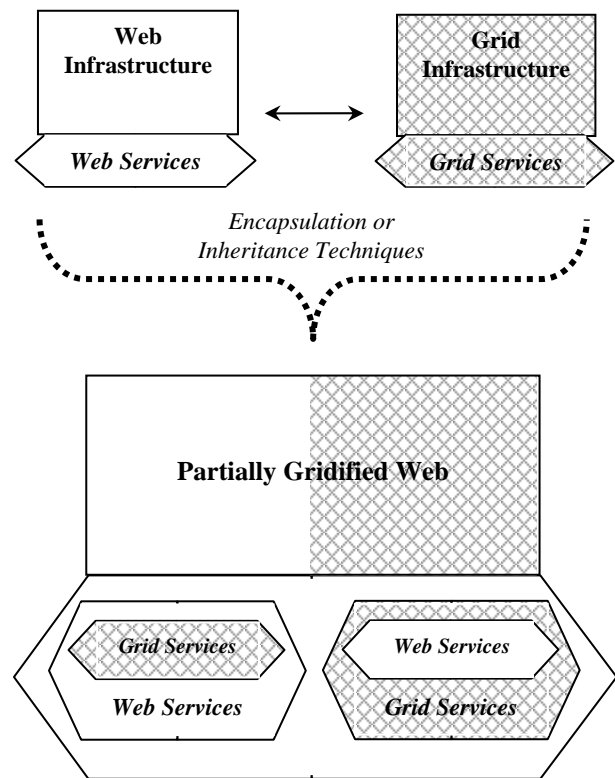


Figure 1. Partially-Gridified Web

A partially-Gridified Web has an infrastructure network that is based on the Web but has a number of small networks or Grid infrastructures built on it. It would provide a platform for carrying out distributed execution and remote processing of any dataset through intensive computation, for storing peta-bytes of massive data and for supporting group-wise collaborative analysis. Users of the partially-Gridified Web resources can be human users, such as computer operators, system administrators, programmers, scientists, or some automated processes or computer programs that send commands/requests to use or discover a Grid resource.

3. GRID SERVICES & WEB SERVICES

3.1 Grid Services as Grid Resources

A resource can be any real or conceptual object that is needed to be accessed by other entities, such as human users of the system or programmes that generate requests for accessing particular resources. The two major categories of Grid resources are physical resources (storage, computational, network and peripheral) and logical (data, knowledge and application) resources. Grid services are a logical type of Grid resources [1] and all the other Grid resources are accessed by using the service type Grid resource. Services can be defined as lines of code that are packaged into a component to perform a dedicated functionality. In terms of their function, services implemented in a network infrastructure can resemble thread-like neuron structures or impulses in a human body. Thus, the output of one service could serve as input to another service. For example, a service can provide a form of data manipulation (storage or retrieval) by using a storage resource, by using extra computational power for calculations via computational resources, by providing solutions to complex problems, or by accessing and using a hardware device, such as a printer, etc. The many social and scientific disciplines, which involve manipulation of massive datasets and need intensive computational power, have the tendency to become Gridified. We can have dedicated Grids possessing specialized functionalities and offering customized services, such as astronomy [11], earthquake simulation [10], and Biomedicine [15], etc. Each of these might have the same management-level services but have different operational-level services. On a Grid infrastructure multiple services are dispersed or scattered across different regions. Their disparate geographical locations, heterogeneous platforms and diverse dynamic statuses make these services rather specialized in nature, and while offering much desired versatility, they are difficult to locate, access and manage. Most importantly, Grid services have changed the vision of Grids from resource-centric to service-centric, and brought forward the widespread adoption of Service Oriented Architecture (SOA) for the implementation of Grid infrastructure.

3.2 Web Services as Communicators

In the new age of technology, Web services are considered as the most efficient and reliable communicators of messages from one place to another regardless of geographical or technological heterogeneity. The emerging Web services provide a framework for an application-to-application interaction that grants access to business-to-business, e-science, and e-government services over the Internet. These services will allow a more extensive use of the Web's functionality by supporting automated processes involving machine-to-machine cooperation and interaction [9]. Thus web services are considered as one of the potential solutions for the practical implementation of Service Oriented Architecture (SOA) in Grid infrastructure. Web services enable interoperability via a set of open standards to provide information about the data in a document to users on various platforms. Web services are built on service-oriented architecture, internet/intranet technologies, and other technologies like information security tools etc. Moreover, Web services support Web-based access, easy integration, and service reusability. With Web service architecture, everything is a service, encapsulating behavior and making it accessible through an interface that can be invoked for use by other services on the network [13]. Since Web technology concepts are independent of any programming language, operating systems platforms or

systems software, they can be easily integrated with other technologies and infrastructures to facilitate intra-technology integration at an advanced level. The traditional Grid applications, with their heavy science focus, typically rely on hosting environments provided by native operating system processes. Grid semantics also usually come from the operating system rather than from external services. Web services, however, can be employed in relatively sophisticated component-based hosting environments, such as Sun's J2EE [8]. Moreover, Web services can communicate with other Web services regardless of their implementation method in order to make them interoperable. Therefore, Web services technologies and components such as XML, WSDL, UDDI or SOAP could be integrated with specialized Grid services or Grid functions to provide enhanced extensible functionality over a partially-Gridified Web.

3.3 Integration of Grid & Web Services

Each of the Grid and Web services are specialized so as to provide sophisticated functionalities in their own domains. Web services can not be directly implemented on Grid architecture due to constraints such as their stateless nature and persistency; whereas the Grid services should always have a state and are transient in nature. Therefore, the integration of Grid services with Web services is necessary in order to support a fully- or partially-Gridified Web. The two types of services have to be glued together using some specialized object-oriented techniques. OGSA defines the Grid service concept, based on principles and technologies from both the Grid computing and Web services communities [9]. Moreover, OGSA not only defines the semantics for a Grid service, but also defines standard mechanisms for creating, naming, and discovering transient Grid service instances. It also provides location transparency and multiple protocol bindings for service instances and supports integration with underlying native platform facilities [12]. Nowadays Grid services are no longer considered separate from the Web services. In fact, according to the OGSF version 1.0 specification [16], a Grid service is considered to be a Web service that conforms to a set of conventions (interfaces and behaviours) which define how a client interacts with a Grid service for such purposes as service lifetime management, inspection, and notification of service state changes [14]. These conventions, and other OGSF mechanisms associated with Grid service creation and discovery, provide for the controlled, fault-resilient, and secure management of the distributed and often long-lived state that is commonly required in advanced distributed applications. Recently there has been a drift from OGSF to Web-Service Resource Framework (WSRF) due to potential performance advantage reasons. WSRF is concerned primarily with the creation, addressing, inspection, and lifetime management of state-enabled resources. It codifies the relationship between Web services and state-enabled resources in terms of the implied resource pattern, which is a set of conventions on Web services technologies. A state-enabled resource that participates in the implied resource pattern is termed a WS-resource. The framework describes the WS-resource definition and association with the description of a Web service interface and describes how to make the properties of a WS-resource accessible through a Web service interface and to manage a WS-resource's lifetime. Based on industry feedback, the revised and updated WSRF specifications are submitted to two new OASIS technical committees, the WS-Resource Framework (WSRF) TC and the WS-Notification (WSN) TC

[17]. All these endeavours are a step forward for the integration of Grid services into Web services.

In this paper, we discuss the two different approaches for integrating Grid and Web services. The first approach suggests that we can build Grid-based Web services, that is, Web services containing special Grid services' interfaces or behaviors (inherited or encapsulated) to make them operable on the Grids. According to the second approach, we can build Web-based Grid services, that is, individual Grid services containing the features & functionalities of the Web services, inherited or extended from the Web services class. Either of these integration approaches could be a good candidate for a viable solution to the Grid resource discovery problem according to their respective implementation environments.

4. GRID RESOURCE DISCOVERY PROBLEMS & METHODS

Grid services provide a means for discovering other Grid resources. The Grid community has not yet been able to fully resolve the issue of Resource Discovery due to various technical constraints and geographical limitations such as autonomous, heterogeneous resources, dynamic nature and status of resources, geographical dispersion of resources, large number of users and large distributed networks, different operating systems/platforms, different administrative domains, lack of portability, availability status of resources and different technology policies. It is not easy to track or locate the right resource in a vast interconnected environment. The mechanism of resource discovery can be viewed through different lenses in various domains; it is a multi-disciplinary task and is one of the most important issues to be dealt with in the future Grid technology. For the successful deployment of a Grid infrastructure, it is essential to access and make maximum use of the resources that are available on the Grid and this is possible only if the resources can be tracked effectively and efficiently. Techniques adopted for resource discovery should be both location and platform independent. When a request is placed for some particular resource, the entire network is first searched to track or locate a suitable resource, and then the resource is matched against the request query and selected (if a sufficient match is established). Upon selection, its availability status is checked and, if available, the desired task is performed. Resource discovery on Grids is quite similar to the discovery of resources on the Web, since in both cases the user is transparent of the specific resource location. This difficulty in tracking down the right resource in a vast interconnectivity environment raises the issue of Resource Discovery, i.e. the process of gaining access to resources for successful completion of the job at hand. For this purpose, a consistent and controlled relationship among the various resources defined on a Grid is needed. As the cost and time taken to complete a job vary tremendously, it is useful to monitor the job consumption rate. Successful allocation, aggregation, management, selection and utilization of autonomous, versatile and distributed resources operating under different authentication policies are issues not resolved yet.

There are many different methods that have been used to address the issue of resource discovery and many approaches have been taken [1], yet a complete solution is not yet available. However, since Web services can be used to discover resources on the Grids, one way of achieving successful resource discovery is by integrating the Grid services with Web services or gluing them

together under the semi-distributed model and using the hybrid approach [24] for resource discovery [24].

5. GRID RESOURCE DISCOVERY USING SERVICE INTEGRATION APPROACHES

The integration of Grid services into Web services might provide a viable solution to the issue of resource discovery. This integration is necessary since Web services cannot directly be used as Grid services due to intrinsic limitations of Web services such as statelessness, etc.; whereas a Grid service must have a state since it is prone to dynamic changes and has more complex functionality than an ordinary Web service. The Grid services could be glued to the Web services for discovering various Grid resources using specialized object-oriented techniques and this can be done using two approaches: either Grid-based Web services or Web-based Grid services, by using encapsulation or/and inheritance techniques and making the new specialized (or "mutant") services available on a partially-Gridified Web. The first approach exploits the inheritance of Grid services into Web services and the second one the inheritance of Web services into Grid services.

Each of these two approaches is suitable for different environments. The first approach is best suited for Web (existing Internet) development/extension with some Grid functionality and the latter is best suited for the development of individual Grids designed to have some sort of Web functionalities. On a partially-Gridified Web (Figure 1), all parts of the Web are not Grid-enabled; rather a small part is operable under Grids, whereas the remaining part is ordinary Web (no Grid functionality). This is one of the reasons as to why this approach is needed.

In order to discover Grid resources, they have to be tracked through some communication means. Web services could play the role of these communication mediums (Section 3.2), via which we can keep track of all the resources available on Grids and facilitate application-to-application interaction. To achieve resource discovery on Grids we have to follow the second, Web-based Grid services approach to define, access, and manage the various Grid resources.

However, studying both integration approaches is essential for the future implementation of a partially-Gridified web on a global scale. Therefore, this marriage of Grid services with Web services is an essential step towards an efficient resource discovery on a partially-Gridified web.

5.1 Grid-Based Web Services

In this approach, the Grid services are inherited (encapsulated or functioned) into the same existing Web service class, instead of having a separate Grid service class (Figure 2). Following this technique would generate Grid-based Web services. The code can be converted into services by specifying each of the interfaces in XML and providing a Web service wrapper [4]. In this way, the Grid services can be embedded into the Web services as dedicated functions in an XML document or we can have specific tags for Grid services. The product would be specialized Grid-based Web services that are dedicated to operate on both the standalone Grid environments as well as on a partially-Gridified Web. [18] presents Web service-based architecture for building specialized Web services and portal clients for Grid applications. The main idea is to wrap Grid application scripts and workflows under a Grid Web service that provides the ability to run instances of the

workflow or script. In many cases, these specialized application services have implementations that are dynamically generated from a simple XML specification by a persistent factory service. Each dynamically generated object (application service or executing application instance) is associated with a resource which is stored for the user. For application execution instances, the resource stores everything needed to recreate the execution including records of execution events and links to the output files.

[19] presents a hybrid composition of Web services and Grid services in order to develop a compositional framework, which brings together Grid services, Web services and Semantic Web technology within a BPEL4WS platform [20]. It is aimed at enhancing BPEL4WS by allowing for the hybrid composition of Web services and Grid services, and by incorporating dynamic binding through agent mediation.

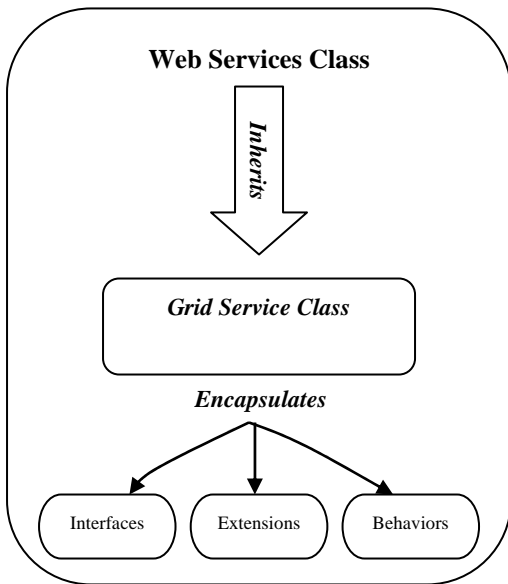


Figure 2. Grid-Based Web Service

[2] uses a Web-based approach and presents a fully decentralized Grid system with graceful scalability named Construction Platform for Specific Computing Grid (CPSCG). CPSCG provides tools to wrap shared computational resources and to export them as web/grid services. Finally, [3] describes how Web services could be developed for Grid applications and used to wrap Grid services. Both of these studies suggest the encapsulation technique for Web-Grid integration.

5.2 Web-Based Grid Services

In this approach, the Web services are inherited (encapsulated or functioned) into the same existing Grid services class (Figure 3). The Grid services can be used to extend the basic Web services by defining a two-layer naming scheme that will enable support for the conventional distributed system transparencies, by requiring a minimum set of functions and data elements that support discovery, and by introducing explicit service creation and lifetime management [5]. Moreover, the Web-based Grid services extend and enhance the capabilities of Web services by providing common and powerful mechanisms that service consumers can rely upon across all services, instead of the service-specific, often ad-hoc approaches typically employed in standard Web services. [21] proposes the creation of state-enabled

resources from conventional HTML Web sites, and the application of hierarchical cluster and wrapper techniques to extract and translate Web resources. It supports service identification and packaging and archives Web site evolution into Grid services environment by exploiting WRSF. It follows the Web-based Grid services approach, since it describes the evolution of Web sites into Grid services environments based on WRSF. [22] presents a high level OGSI-compliant hierarchical Information Service built on Globus Toolkit MDS-3. Following the OGSA standard, it integrates with local information suppliers that are implemented as OGSI-compliant Grid services, such as local resource management systems, job Grid service, job queuing Grid service, etc., and supports information collection, update and accessing on a Grid Virtual Organization that consists of multiple administrative domains and resources.

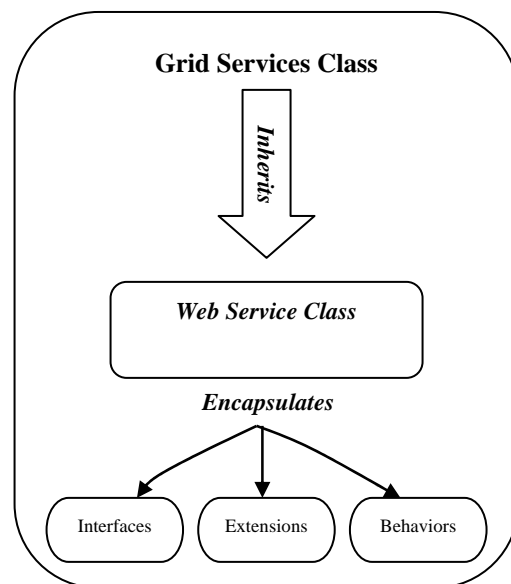


Figure 3. Web-Based Grid Service

The proposed information model includes job status, computational resources, local resource workload, service metadata, and queue status. This information is further classified into two categories, the static information and the dynamic information.

The stateless nature of Web services, which is a limitation and creates a need for integration, can be addressed by having a separate Grid class that is inherited by the Web class. It will have all the functionalities of a typical Web service and also have some additional functionalities, such as service state, service interfaces, service structure, etc.

[23] presents the evolution of Web-services compliant TACWeb, which is a web-based Grid portal built on top of the Globus Toolkit. It is an interactive environment that deals with complex user requests, regarding the acquisition of biomedical data and the processing and delivering of biomedical images, using the power and security of Grids. The basic capabilities are encapsulated and exposed as Web Services, allowing the development of new health applications as compositions of such services. The use of Grid portal technology is proposed for the development of user-centred applications, such as an Electronic Patient Record system to support the management of biomedical images.

5.3 Grid-based Web versus Web-based Grid

There is a lot of on-going work exploring solutions for successful resource discovery in Grid infrastructures. At the meeting of the Global Grid Forum (GGF) held in March 2004, a proposal for an evolution of OGSi based on a set of new Web Services specifications - WS-Resource Framework (WS-RF) and WS-Notification (WS-N) - was presented [7]. It supports the extension of Grid services from Web services, which is the Web-based Grid services approach using the inheritance technique of integration. Today most of the ongoing work involves using the WS-RF for the extension and integration of Grid services into Web services.

Employing the Grid-based Web service approach for Grid-Web integration would be more complicated to implement (this implementation has to take place in a purely Grid environment), as the extension of dedicated functions defined for the Grid services class into the Web services class can be a task of high complexity. On the other hand, adoption of the Web-based Grid approach in a purely Grid environment may be more effective because in this way we have the entire Web service class packaged and integrated into the Grid service class and each Grid service object, thus instantiated, would have all the capabilities of a Web service and some additional specialized capabilities of its own.

5.4 SOAP-RD: An Integration Example

An example of integrating Grid services and Web services is discussed here, in which the SOAP Web service technology is fused with an instance of the Grid Resource Discovery (RD) service.

The standard Simple Object Access Protocol (SOAP) was developed to enable the transmission and reception of messages [25] for accessing distributed resources or objects. The Grid-specific Resource Discovery (RD) service, which is one of the most important operations/services on Grids, is used not only to access but also to track, match, select and request [1] the geographically distributed resources or objects. This Grid RD service is a set of many sub-services or instances, and one of its instances can be glued to the SOAP Web service for the simplification of the larger resource discovery problem.

Following the first approach of Grid-based Web service, it would be essential to rewrite the original SOAP class so that it extends (inherits) the functionalities of the Grid RD service class since, being merely a Web service, SOAP has limited functionalities. It would be just like re-engineering the entire protocol, which is an error prone and complicated process.

Whereas, following the second approach of Web-based Grid service, it is not needed to modify or rewrite the original SOAP class (re-engineering is not required), rather this class needs to be simply extended or inherited by a Grid-specific/scripted RD service class performing the basic SOAP operations and possessing extra capabilities that are Grid-specialized (such as service state, etc.). This solution supports effectively support the operation of resource discovery in Grid environments or Partially-Gridified Web. Furthermore, it is relatively simple and less prone to errors.

Therefore, instead of re-creating (re-scripting) the existing Web services to include new Grid functionalities, it is advisable to

develop new Grid services and include (encapsulate) in them the existing Web service functionalities.

A further reason why it is beneficial to use the existing web infrastructure and web services technologies as the standard platform for the deployment of Grid infrastructure is that is beneficial since the Web Services have presently reached to a state of high standardization state, whereas Grid technology, being a relatively newly emerged discipline, has no infrastructural standards and benchmarks established yet. Thus it seems that it is by and large easier to use the existing standardized paradigm of Web Services which is widely established and with a long history of successfully implementations.

6. CONCLUSIONS

The designing and deployment of a Grid infrastructure is the subject of a relatively new area, which has not yet reached at such the maturity stage of effective standardization. Many of the problems associated with the implementation of Grids could be resolved if the Web is used as the underlying ("backbone") infrastructure for the development of Grids, thus resulting in the formation of a Grid-enabled or a partially Gridified-Web.

In this paper, we argue that Grids should be developed using the underlying Web infrastructure and Grid services could be integrated with Web services using inheritance or encapsulation techniques to produce Grid-based Web services.

The paper presented Grid services as one type of Grid resources, explored possible ways to integrate Grid services and Web services, and discussed how this will support Grid resource discovery. The two approaches, namely the Grid-based Web service approach and the Web-based Grid service approach, are examined critically, and suggestions are made as to their suitability for implementation in their respective environments. On a partially-Gridified Web, the use the Web-based Grid services approach seems to be more directly applicable and more appropriate for the integration of Grid services and Web services and for addressing the issue of resource discovery.

7. REFERENCES

- [1] Naseer, A., and Stergioulas, L.K. Resource Discovery in Computational Grids: State-of-the-art and Current Challenges. *Self-Organization and Autonomic Informatics (I)*, IOS Press, Amsterdam, Netherlands, pp. 237-245, 2005.
- [2] Zheng, W., Hu, M., Yang, G., Wu, Y., Chen, M., Ma, R., Liu, S., and Zhang, B. An Efficient Web Services Based Approach to Computing Grid. *Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04)*, 2004.
- [3] Yang, X., Hayes, M., and Usher, A. Developing Web Services in a Computational Grid Environment. *Proceedings of the 2004 IEEE Conference on Service Computing (SCC'04)*, 2004.
- [4] Fox, G. Grids of Grids of Simple Services. *Computing in Science and Engineering*, 2004.
- [5] Grimshaw, A., and Tuecke, S. Grid services extend Web services. *Web Services Journal*, 3(8):22-26, 2003.

- [6] Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., and Vambenepe, W. The WS-Resource Framework. March 5, 2004.
- [7] Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Maguire, T., Snelling, D., and Tuecke, S. From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution. March 5, 2004.
- [8] Goth, G. Grid Services Architecture Plan Gaining Momentum. *IEEE Internet Computing*, 6(4):11-12, Jul-Aug 2002.
- [9] Talia, D. The Open Grid Services Architecture: Where the Grid Meets the Web. *IEEE Internet Computing*, 6(6):67-71, Nov-Dec 2002.
- [10] Peng, J. and Law, K. Reference NEESgrid Data Model, tech. report, Network for Earthquake Engineering Simulation Grid (NEESgrid), NEESgrid-2004-40, 2004.
- [11] Qin, C.L. et al., "Myspace: Personalized Work Space in Astrogrid," Proc. UK E-Science All Hands Meeting, University of Southampton, pp. 361-365, 2003. www.allhands.org/uk/2003/proceedings.
- [12] Foster I., Kesselman, C., Nick, J. M., and Tuecke, S. Grid Services for Distributed Systems Integration. *IEEE Computer* 35(6):37-46, Jun 2002.
- [13] Wu, F.F., Moslehi, K., Bose, A. Power System Control Centers: Past, Present, and Future. *Proceedings of the IEEE* 93(11):1890-1908, Nov 2005.
- [14] Foster, I., Czajkowski, K., Ferguson, D.E., Frey, J., Graham, S., Maguire, T., Snelling, D., Tuecke, S. Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. *Proceedings of the IEEE* 93(3):604-612, Mar 2005.
- [15] C.A. Stewart, "Bioinformatics: Transforming Biomedical Research and Medical Care". *Communications of the ACM*, 47(11):30-33, 2004.
- [16] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Snelling, D., and Vanderbilt, P. (2003) Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum, www.ggf.org.
- [17] Baker, M., Apon, A., Ferner, C., Brown, J. Emerging Grid Standards. 38(4):43-50, April 2005.
- [18] Gannon, D., Alameda, J., Chipara, O., Christie, M.; Dukle, V., Fang, L., Farrellee, M., Kandaswamy, G., Kodeboyina, D., Krishnan, S., Moad, C., Pierce, M., Plale, B., Rossi, A., Simmhan, Y., Sarangi, A., Slominski, A., Shirasuna, S., Thomas, T. Building Grid Portal Applications from a Web Service Component Architecture. *Proceedings of the IEEE*, 93(3):551-563, March 2005.
- [19] Anane, R., Chao, K.-M., Li, Y. Hybrid Composition of Web Services and Grid Services. *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*, Mar-Apr 2005, pp. 426-431.
- [20] Khalaf, R., Mukhi, N. and Weerawarana, S., Service-oriented Composition in BPEL4WS, http://www2003.org/cdrom/papers/alternate/P768/choreo_html/p768-khalaf.htm, 2003.
- [21] Li, J., Yang, H. Towards Evolving Web Sites into Grid Services Environment. *Seventh IEEE International Symposium on Web Site Evolution*, 2005. (WSE 2005). Sept. 2005, pp. 95-102.
- [22] Zang, T., Jie, W., Hung, T., Lei, Z., Turner, S.J., Cai, W. The Design and Implementation of An OGSA-based Grid Information Service. *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*.
- [23] Aloisio, G., Blasi, E., Cafaro, M., Fiore, S., Lezzi, D., Mirto, M. Web Services for a Biomedical Imaging Portal. *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03)*.
- [24] Naseer, A., and Stergioulas, L.K. Resource Discovery in Grids and Other Distributed Environments: States of the Art. To appear in *Multiagent and Grid Systems - An International Journal*, IOS Press, ISSN: 1574-1702, 2006.
- [25] Twardoch, A. Web Services Technologies: XML, SOAP and UDDI. .Net and Web Services Seminar, European University Viadrina Frankfurt (Oder), Germany, <http://www.twardoch.com/webservices>, 2003.