Should We Compare Web and Grid Services?

Jyotishman Pathak[⋆]
Artificial Intelligence Research Lab
Center for Computational Intelligence, Learning and Discovery
Department of Computer Science, Iowa State University, Ames, IA 50011, USA
jpathak@cs.iastate.edu

Abstract. Earlier work in the areas of Service-Oriented Architecture (SOA) and Grid computing resulted in the Open Grid Services Infrastructure (OGSI) specification, which defined Grid services as an extension of Web services. As a result, people often try to understand the relationship between Grid and Web services or even compare between them. This paper clarifies that, with the on-going developments in Web services architecture (e.g., WS-Resource Framework), it is inappropriate to consider Grid services extending Web services, and hence, distinguish between them. As of today, the term 'Grid service' has been deprecated by the Open Grid Services Architecture, and has no technical significance.

1 Introduction

The advent of the *Web services* technology [1], in the year 2000, provided a new distributed computing paradigm that differed from the previous approaches (e.g., CORBA, Java RMI) by relying on Internet-based standards (XML [3], SOAP [10], WSDL [4]) to address implementation of heterogeneous and loosely-coupled distributed applications. It allows designing of software systems to support interoperable machine-to-machine interaction over a network. However, the Web services architecture lacked in providing the notion of state and stateful interactions, resource lifecycle management, notification of state changes, and support for sharing and coordinated use of diverse resources in dynamic 'virtual organizations' [6]—issues that are of central concern to the developers of distributed systems (e.g., large-scale scientific research). To address these problems, the OGSI specification [11] introduced *Grid services*—Web services that provide a set of well-defined interfaces and follows specific conventions. It introduced the idea of a *stateful* Web service and defined approaches for creating, naming, and managing lifetime of instances of a service; for declaring and inspecting service state data; for asynchronous notification of service state change; for representing and managing collections of service instances; and for common handling of service invocation faults. One of the major purposes of this specification was to enable Grid services to represent and virtualize Grid resources such as physical devices, software components, and executing jobs, to support a collaborative computing environment.

However, this approach was not widely accepted by the Web services community, and concerns were raised about the significance of Grid services and the relationship between Grid and Web services. To resolve these issues, lately there has been some collaboration between Web services and Grid computing community [5] towards refactoring and evolution of Grid standards aimed at aligning OGSI functions with emerging consensus on Web services architecture [1]. This effort has produced two important sets of specifications: WS-Resource Framework [9] and WS-Notification [8], which essentially retain all the functional capabilities present in OGSI, and at the same time build

[⋆] This work was done while the author was an intern at Hewlett-Packard.

on broadly adopted Web services concepts. As a result of this refactoring, the term 'Grid service' was deprecated and hence, it is inappropriate to consider Grid services as extension of basic Web services. In this paper, we try to clarify this specific point that distinguishing Web and Grid services has no significance. We begin by briefly introducing the Web services technology in the next section.

2 Web Services

The W3C Web services architecture working group [1] defines Web service as a software system designed to support interoperable machine-to-machine interaction over a network. A Web service provides an interface described using a machine-processable WSDL [4] document, and other systems can interact with the service in a manner prescribed by its description using SOAP [10] messages. Typically, the WSDL document defines message formats, datatypes, transport protocols, and transport serialization formats that should be used between the service requester and service provider. It also specifies one or more network location(s) at which a service provider can be invoked, and may provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service.

At large, Web services are regarded as *stateless* and *non-transient*. *Statelessness* means that a Web service does not remember information, or keep *state*, from one invocation to another. Even though some Web service interfaces frequently provide a user with the ability to access and manipulate state (e.g., values in a database), the notion of stateful resources acted upon by the Web service implementation remains 'implicit' in the interface definition [6]. A Web service is considered *non-transient*[1] as it does not have the concept of service creation and destruction, and hence managing its lifecycle. However, the use of state, transient or persistent, is critical to the operation of many Web services, and important in building distributed e-Science and e-Business applications. This drawback was one of the main motivations for the Grid community to introduce the notion of 'Grid services', which we discuss next, as an extension of Web services.

3 Grid Services

Based on the Web services technology, the OGSI [11] specification defines a Grid service to be a Web service that conforms to a set of conventions (interfaces and behaviors) that define how a client interacts with a Grid service. They solve the *stateless* and *non-transient* problems of Web services by introducing a *factory/instance* model. Instead of having stateless services shared by many clients, a Grid service *factory* is used to create and maintain multiple *instances* of the Grid service, each representing one resource. Typically, it is possible to have one-to-many, many-to-one, and many-to-many interactions between clients and instances of Grid services. These instances are transient in nature, as opposed to being bound to the lifetime of the Grid service's container. As a result, the instances have limited lifetime which can be managed by the client itself. The states are maintained separately by the service instances at the service provider site. In

[1] A *transient* service instance is one that can be created and destroyed. Usually, they are created for specific clients and do not outlive their clients.

addition, a Grid service can also be configured to be a 'notification source', and certain clients to be 'notification sinks' (or subscribers). This means that if a change occurs in the Grid Service, that change is notified to all the subscribers.

However, even though Grid services addressed problems related to statefulness, transience, lifetime management, state change notifications, etc., they were not widely adopted by the Web services community and questions were raised about the significance of Grid services. One of the main reasons was because OGSI modeled a stateful resource as a Web service that encapsulates the resource's state, with the identity and lifecycle of the service and state coupled. This was a serious concern to the Web services pundits who argued Web services should be stateless. Furthermore, the specification did not have a clean separation of functions to support incremental adoption [6]. Also, due to aggressive use of XML Schema (e.g., `xsd:any` attributes) and document-oriented WSDL operations, the specification did not work well with existing Web services and XML tooling. Thus, in a way, OGSI encouraged a major part of the Web services industry to pursue a model that violated the conceptual loose coupling of Web services-based SOA. To resolve these issues, the Web services and Grid computing communities collaborated for refactoring and evolution of OGSI targeted at aligning its functions with the emerging Web services standards [1]. This resulted in proposing a set of new Web services specifications—*WS-Resource Framework (WSRF)* [9] and *WS-Notification (WSN)* [8]—which have been adopted as the basis for Open Grid Services Architecture (OGSA) [7] based Grids. Subsequently, the term 'Grid service' was deprecated by the OGSA, and as a consequence, from the OGSA perspective the term has no technical significance and its usage is discouraged.

4 WS-Resource Framework & WS-Notification

WSRF [9] expresses state as stateful resources and specifies the relationship between Web services and stateful resources in terms of the WS-Resource Access Pattern, a set of conventions based on Web services standards (e.g., XML [3], WSDL [4], WS-Addressing [2]).

In general, a WS-Resource is defined as a Web service through which a resource can be accessed. It is required to have a specific set of state data expressible as an XML document, a well-defined lifetime, and to be known to, and acted upon by, one or more clients. This data is defined by the Resource Properties document schema, which is referenced by the WSDL description of the service, and which explicitly describes a view of the stateful resource with which the client interacts. The location of a WS-Resource is described using a WS-Addressing [2] Endpoint Reference (EPR). The EPR comprises a reference to the stateless Web service and an XML serialization of the stateful resource identifier. WSRF avoids the need to describe the identifier explicitly in the WSDL description. Instead, the identifier is encapsulated within the EPR and implicitly included in all messages addressed through the EPR according to the rules of WS-Addressing. In addition, WSRF comprises various specifications for managing properties of resources (*WS-ResourceProperties*), resource lifetime management (*WS-ResourceLifetime*), creating heterogeneous collections of services (*WS-ServiceGroup*), and commonly interoperable mechanisms for fault handling (*WS-BaseFault*).

Clearly, in an environment where resources may be created/destroyed or may change their state dynamically, it is vital to provide the functionality of asynchronously noti-

fying the client/consumer about state changes. To address this, the WSN [8] family of specifications was introduced for providing a generic, hierarchical topic-based approach for notification using the traditional publish-subscribe model. The WS-Resource properties define a mapping from element names of resource properties to topic names (items of interest for subscription). Typically, a service publishes a set of topics that clients can subscribe to, and receive notifications whenever the topic changes.

5 Conclusion

In this paper, we try to clarify the ambiguity between Grid and Web services technologies. Earlier, Grid services were considered as an extension of Web services, by introducing statefulness and transience. However, due to lack of clear factoring about functionalities of Grid services and non-compliance with commonly used Web services and XML tooling, the OGSI specifications were not widely accepted by the industry and people often began to question the significance of Grid services. As a result, a refactoring of OGSI was proposed which led to the development of a set of specifications, particularly, WS-Resource Framework and WS-Notification, which align with the functionalities specified originally as part of OGSI and at the same time build on emerging Web services standards. As a consequence of this refactoring, the term 'Grid service' was deprecated by OGSA, and hence, from the OGSA standpoint the term has no technical meaning and we should avoid using it.

Acknowledgment. The author would like to thank Jem Treadwell and Latha Srinivasan from Hewlett-Packard for their contributions and helpful discussions.

References

1. D. Booth, H. Haas, F. McCabe, and et al. Web Services Architecture, W3C Working Group Note 11 February 2004. Available at *http://www.w3.org/TR/ws-arch/*, 2004.
2. D. Box, E. Christensen, F. Curbera, and et al. Web Services Addressing, W3C Member Submission 10 August 2004, Available at *http://www.w3.org/Submission/ws-addressing*, 2004.
3. T. Bray, J. Paoli, and et al. Extensible Markup Language, Version 1.1. Available at *http://www.w3.org/TR/2004/REC-xml11-20040204/*, 2004.
4. E. Christensen, F. Curbera, and et al. Web Services Description Language, Version 1.1. Available at *http://www.w3.org/TR/WSDL/*, 2001.
5. K. Czajkowski, D. Ferguson, I. Foster, and et al. From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution, version 1.1. Available at *http://www.globus.org/wsrf/specs/ogsi_to_wsrf_1.0.pdf*, 2004.
6. I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Available at *http://www.globus.org/alliance/publications/papers/ogsa.pdf*, 2002.
7. I. Foster, H. Kishimoto, A. Savva, D. Berry, and et al. The Open Grid Services Archictecture, Version 1.0. Available at *http://forge.gridforum.org/projects/ogsa-wg*, 2005.
8. S. Graham, D. Hull, B. Muray, and et al. Web Services Notification, Version 1.3 . Available at *http://www.oasis-open.org/committees/wsn*, 2005.
9. S. Graham, I. Robinson, and et al. Web Services Resource Framework Primer, Draft 5. Available at *http://www.oasis-open.org/committees/wsrf*, 2005.
10. M. Gudin, M. Hadley, and et al. Simple Object Access Protocol, Version 1.1. Available at *http://www.w3.org/TR/soap12/*, 2003.
11. S. Tuecke, K. Czajkowski, I. Foster, and et al. Open Grid Services Infrastructure, Version 1.0. Available at *http://forge.gridforum.org/projects/ogsi-wg*, 2003.