Abstract

The specifications produced by the Database Access and Integration Services (DAIS) working group prior to GGF11 describe mappings onto the Open Grid Services Infrastructure (OGSI) v1.0. The group of specifications collectively referred as the Web Services Resource Framework (WSRF) has recently been proposed as a replacement for OGSI. Furthermore, the Web Services Grid Application Framework (WS-GAF) is a proposal of patterns that utilise existing Web Services technologies for building Grid applications. WS-GAF also considers the emerging WS-Context specification. In light of these the DAIS working group is considering which mappings should be presented in future versions of its specifications. This document summarises the result of some scenario based investigations into a number of possible mappings.

1. Introduction

The Global Grid Forum (GGF) Database Access and Integration Services (DAIS) [1] working group is defining the concepts and design patterns necessary for working with data in a Grid environment. Until recently, the Open Grid Services Infrastructure (OGSI) v1.0 [2], which defined fundamental concepts and characteristics for Grid Services, was considered as the infrastructure of choice for DAIS.

In August 2003, a group of authors from the University of Newcastle and Arjuna presented a “Grid Application Framework based on Web Service Specifications and Practices” (WS-GAF) [3]. This proposes an infrastructure for grid services based on the specifications that are part of the Web Services Interoperability (WS-I) Profile 1.0a [4] and WS-Context [5]. In more recent work WS-GAF proposes the use of URNs [6] for the identification of resources.

In January 2004, a group of authors from IBM and the Globus Alliance announced the Web Services Resource Framework (WS-RF) [7] as “the convergence of Web and Grid Services” and the evolution of OGSI. Many of the features that OGSI introduced, like Grid Service Instances, Grid Service Handles and Service Data Elements were reworked, so addressing many of the concerns expressed in [3] and the Web Services community in general.

Given that there are different possible mappings of the DAIS concepts to underlying infrastructure solutions, the DAIS working group decided to investigate the alternatives and identify the issues that may arise in a number of representative cases. The GGF document “Scenarios for Mapping DAIS Concepts” describes the findings of this investigation. This paper summarises those findings for the UK eScience All Hands 2004 meeting.

2. Goals of the Investigation

The investigation into the potential choices of infrastructure for mapping the DAIS concepts aims to identify possible issues that relate to:

1. The nature of the interactions with data services (the message formats)
2. The representation of data resources through service interfaces (the realisations)
3. Accessing data resources from
   a. Different portTypes offered by the same service
   b. portTypes offered by different services
4. Providing names for data resources
5. Creation/generation of data resources through interactions (factory patterns)
6. Access to metadata about data resources and/or data services
7. Stateful interactions with data resources (e.g. interacting with a result set)
8. Stateful interactions with data services (e.g. setting a logging context)
9. Locating data resources (registries)
10. Third-party delivery of data resources

Other issues that are important but are not considered by this investigation include:
1. Orthogonal functionality (security, policies, transactions, coordination, reliable messaging, orchestration, etc.)
2. Notification
3. Representation of data resources outside the boundaries of services (data formats)

3. Approach

The investigation concentrates on a number of scenarios in which consumers and services interact:
1. Stateful interactions with data resources
2. Sessions with data services
3. Discovery of data resources
4. Access to metadata about data resources and/or services
5. Third-party delivery

Three different underlying infrastructures are assumed for mapping the DAIS concepts:
- Web Services-Interoperability (WS-I) only specifications
- WS-I plus WS-Context
- WS-I plus WSRF

The way in which each of the above infrastructures can support each of the scenarios is examined and the differences or commonalities are identified.

4. Example Scenario

One scenario is presented here as an example of the approach taken. The reader is referred to the original document [8] for information on all of the scenarios considered to date.

4.1. Scenario 1 - Stateful interactions with data resources

A consumer discovers a data service from a registry (out of scope). This service provides access to some data resource of interest.

The consumer submits a query to the service.

1. The result (dataset) may be returned as part of the response.
2. The result may be kept at the service, as a data resource, and only a name of the data resource is returned. The name can be the actual name of the data resource or, most likely, a logical name.

In case 2, the consumer may wish to correlate subsequent messages to the service with the identified data resource (e.g., a sort operation). The three approaches considered are:
- Use of the resource name (the resource name is carried as part of the message’s body)
- Contextualisation using WS-Context (using a context managed outside the consumer and data service)
- Using WSRF

4.2. Contextualisation using WS-I stack

Logical names (e.g., URIs, URNs, or other application-domain specific identifiers) are used to identify resources outside the boundaries of a service. The messages are contextualised (correlated) through the inclusion of logical resource names in the body of each message.

In Figure 1, a query is sent to a Web Service and the (logical) name of a data resource is returned.
has to maintain a mapping (this is true for all the examples in this document).

Figure 2: A message carrying the name of the data resource in the body of the message

Figure 2 shows another interaction with the data service of Figure 1. The message sent to the service is logically correlated with the data resource produced by the previous message by including the name of the data resource.

Figure 3: Interactions with multiple services using the same (logical) name for the data resource

If the data resource is subsequently replicated, for example, for quality of service reasons (through some infrastructure that is out of scope here), and access to it is offered by another service, then explicit contextualisation can still be used assuming that the same (logical) name is maintained (Figure 3).

The identity can also be passed off to a third party which can use it to access the data in the same way as the original consumer.

4.3. Contextualisation using WS-Context

In this example, an infrastructure supporting contextualisation is assumed (e.g., WS-Context) which models stateful interactions between services or between consumers and services.

A context structure is created (out of scope for this example) and is used in all the related message exchanges. A message sent to the service results in the identified data resource being associated with the stateful interaction. The context structure could be maintained by the participants in the interaction or by a separate service, according to the semantics of the specification used.

Figure 4: A data resource is generated within the scope of a contextualised interaction

A consumer sends a query to a service within the scope of a contextualised interaction (Figure 4). The service makes an association between the context and the resulting data resource. No name needs to be exposed and returned as part of the message interaction. It is assumed that the consumer somehow indicates to the runtime what context to use (how this is done is out of scope for this document) in subsequent messages.

Figure 5: A contextualised message that is associated with the resulting data resource

In Figure 5, the consumer sends a contextualised message to the service. The service interprets that message within the context of the identified interaction.

Figure 6: Messages sent to multiple services within the context of the same interaction
As shown in Figure 6, the same context can be used to send messages to multiple services that have correlated the same dataset with the context that identifies the interaction.

Another pattern of using context is shown in Figure 7 where a (logical) name for a dataset is known to the consumer (e.g., it was returned to the consumer as shown in Figure 1). The consumer associates the identified dataset with the context of the interaction by sending an appropriate message to the service. Subsequent messages within the scope of the same interaction are interpreted by the service as being correlated with the identified dataset, because the dataset ID/name is implicitly sent across in the context.

WS-Addressing ensures that the ReferenceProperties element of the WS-Addressing construct is included in every subsequent interaction with the service that relates to the generated data resource, as shown in Figure 9.

Figure 7: A dataset is associated with the context of an interaction

4.4. Contextualisation using WS-RF

In WS-RF, a resource, like a data resource, is identified through a WS-Addressing [9] Endpoint Reference (EPR) construct. The EPR contains endpoint-related information about a service and data specific to the service about the identified resource.

In Figure 8, a message with a query is sent to a service and an EPR identifying the resulting data resource is returned.

Figure 8: A query is sent to a Web Service and a WS-Addressing construct is returned

Figure 9: Implicit contextualisation using the contents of the WS-Addressing ReferenceProperties element

The WS-RF specification defines that the contents of the ReferenceProperties element are opaque to the consumer and they should not be interpreted in any way in order to reason about the identity of the dataset.

If a copy of the same data resource is to be accessed via a different service, a new EPR must be obtained.

If the data resource is to take part in a stateful interaction that involves multiple participants, then an approach is to explicitly associate the identity of the data resource with an externally managed context (as in the example of Figure 7).

5. Summary of Findings

This section refers back to the points listed in section “Goals of the Investigation” and notes observations made while considering the various approaches.

5.1. The nature of the interactions with data services

- **WS-I**: If a resource must be identified, its identity must be included in the body of a message.
- **WS-Context**: Identity can be sent using WS-Context in the headers, without having to declare it as a parameter.
- **WS-RF**: The reference property element in an EPR identifies a WS resource in a web service. Knowledge of format of reference properties not required by consumer

5.2. The representation of data resources through service interfaces

- **WS-I**: Services with appropriate interfaces for a data resource are located via registry or out of band mechanism. Data resources identified using application-domain
defined conventions and identifier carried in application payload of a message.

- **WS-Context**: Services with appropriate interfaces for a data resource located via registry or out of band mechanism. The Data resource is identified using a WS-context that is passed as part of the message header. The identity is assumed to be declared, as part of the metadata of the service, for example.

- **WS-RF**: Services with appropriate interfaces for a data resource located via registry or out of band mechanism. Data resource identified as part of the EPR obtained. EPR resolves to information passed as part of the message header.

### 5.3. Accessing data resources via different port-Types offered by the same service

There is no difference across approaches. PortTypes are aggregated into the one service and resource is identified in accordance with the approach in question.

### 5.4. Accessing data resources via different port-Types offered by different services

- **WS-I**: Resource identity must be available and valid across services. Resource identity used with one service is passed in with message bodies to other services that are able to interpret it. Suitable services must be located via a registry or via out of band mechanisms. Resource identity must be available and valid across services.

- **WS-Context**: Context used for one service is passed in with message headers to other services that are able to interpret it. Suitable services must be located via a registry or via out of band mechanisms. Requires that the data resource identity is available across services.

- **WS-RF**: In the implied resource pattern an EPR identifies a specific ws resource. To access that resource through another web service would involved retrieving a new EPR from the existing service, from a registry or via an out of band mechanism.

### 5.5. Providing names for data resources

- **WS-I**: The name and the information used to identify the data resource can be one and the same. Other metadata can be associated with this name in a registry.

- **WS-Context**: The name and the information in the context used to identify the data resource can be one and the same. Other metadata can be associated with this name and/or context in a registry.

- **WS-RF**: Reference properties are designed to be opaque and are distinct from the name (or other metadata) that a consumer would use to locate a data resource via a registry.

### 5.6. Creation/generation of data resources through interactions (factory patterns)

- **WS-I**: Such a process may result in a new resource identifier. To use the new identifier a suitable service endpoint must also be returned by the factory operation or must be located via a registry or via out of band mechanisms.

- **WS-Context**: Such a process may result in a new resource identifier. To use the new identifier a suitable service endpoint must also be returned by the factory operation or a suitable service must be located via a registry or via out of band mechanisms. The identifier is passed in the message header via a WS-Context.

- **WS-RF**: Such a process would result in a new EPR. The new EPR can be used directly to access the new resource.

### 5.7. Access to metadata about data resources

- **WS-I**: A service must be designed and implemented to provide access to metadata.

- **WS-Context**: A service must be designed and implemented to provide access to metadata.

- **WS-RF**: WS-ResourceProperties is a proposal for providing access to metadata using the EPR to identify the data resource.

### 5.8. Access to metadata about data services

No difference across approaches. A service must be designed and implemented to provide access to metadata.

### 5.9. Stateful interactions with data resources (e.g. interacting with a result set)

- **WS-I**: Not explicitly part of the model. Identity of the resource can be passed in the body.

- **WS-Context**: Not explicitly part of the model. Identity of the resource can be passed using a context in the header or the body.

- **WS-RF**: The implied resource pattern is used identify the resource.

### 5.10. Stateful interactions with data services (e.g. setting a logging context or establishing a session)

- **WS-I**: Not explicitly part of the model. Identity can be passed in the body.
• **WS-Context**: Not explicitly part of the model. Identity can be passed using a context in the header or the body.
• **WS-RF**: Can use either a service specific mechanism or something like WS-Context.

5.11. Locating data resources (registries)

• **WS-I**: Metadata held by a registry allows a consumer to identify the endpoint of the service able to act on a named resource.
• **WS-Context**: Metadata held by a registry allows a consumer to identify the endpoint of the service able to act on a named resource.
• **WS-RF**: Metadata held by a registry allows a consumer to identify the EPR of a WS resource able to act on a named resource.

5.12. Third-party delivery of data resources. Assumes that data to be delivered is identified by a resource identifier

• **WS-I**: The resource identifier along with the service endpoint is passed to the third party which arranges delivery. Third party could look up service in a registry if necessary but not required. Assumes that data to be delivered is identified by a resource identifier.
• **WS-Context**: The resource identifier along with the service URL is passed to the third party, using a WS-Context, which arranges delivery. Third party could look up service in a registry.
• **WS-RF**: Assumes that data to be delivered is identified by an EPR. The EPR is passed to the third party which arranges delivery.

5.13. Deployment

• **WS-I**: Any web service container, e.g. Tomcat/axis, WebSphere/Oracle Application Server 10g.
• **WS-Context**: Any web service container + WS-Context implementation.
• **WS-RF**: Any web service container + code to manage addressing, properties, lifetime. E.g GT4 (when available).

5.14. Lifetime management of resources

• **WS-I**: Interfaces must be defined to manage the lifetime of resource. Specifically the artifacts associated with resource identities.
• **WS-Context**: Interfaces must be defined to manage the lifetime of resource. Specifically the artifacts associated with resource identities.
• **WS-RF**: WS-ResourceLifetime is a proposal for the management of resource lifetimes.

5.15. Security

Not specifically addressed but no issues specific to approaches anticipated.

6. Observations

In this section we attempt to identify any possible implementation issues with adopting one approach over the other with regards to specific issues.

6.1. Stability of supporting infrastructure

• **WS-I**: The specifications that are part of the WS-I Profile 1.0a (SOAP, WSDL, UDDI) are considered to be stable.
• **WS-Context**: The WS-Context specification is part of the WS-CAF [10] suite of specifications which has been in the OASIS standardisation process since August 2003.
• **WS-RF**: The WS-RF suite of specifications has been in the OASIS standardisation process since March 2004.

6.2. Interoperability and adoption

• **WS-I**: All major Web Services vendors support the specifications in the WS-I profile and there interoperability tests from the WS-I organisation.
• **WS-Context**: WS-CAF is supported by Oracle/Sun/IONA and other companies but not by IBM and Microsoft. There are no available implementations to test interoperability.
• **WS-RF**: The WS-RF suite of specifications is supported by IBM/HP/Globus and others but not by Microsoft/Oracle/Sun. Private interoperability tests between implementations have taken place.

6.3. Composability

• **WS-I**: There are no issues with using other, existing WS specifications.
• **WS-Context**: There are no issues with using other, existing WS specifications.
• **WS-RF**: Issues with using BPEL and WS-TransactionManagement (part of WS-CAF).
6.4. Tooling

- **WS-I**: Widely available tooling (commercial and open source).
- **WS-Context**: No available tooling yet – Vendors supporting WS-Context are in the process of implementing necessary tools.
- **WS-RF**: Experimental implementations emerging.

6.5. Conceptual Model

- **WS-I**: Resources are not explicitly modelled the focus is on services.
- **WS-Context**: Resources are not explicitly modelled the focus is on services.
- **WSRF**: Explicitly models resources and interactions with them.

7. Trademarks

IBM, DB2 and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Oracle is a trademark of Oracle Corporation.

Globus Project and Globus Tookit are trademarks of the University of Chicago.

8. Copyright Notice

Copyright (C) Global Grid Forum (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

9. References