GWD-R-P.200
OGF DAIS Working Group

Isao Kojima, AIST
Said Mirza Pahlevi, AIST
Steven Lynden, AIST

10 January 2013

Web Services Data Access and Integration – The RDF(S) Realization: WS-DAI-RDF(S) Querying Specification, Version 1.0

Status of This Memo

This memo provides information regarding the specification of service-based interfaces to RDF data resources. Distribution is unlimited.

Abstract

Data resources play a significant role in many applications across multiple domains. Web services provide implementation neutral facilities for describing, invoking and orchestrating collections of networked resources. The OGF (Open Grid Forum) Open Grid Services Architecture (OGSA), and its associated specifications, define consistent interfaces through web services to components of the grid infrastructure. Both the web and grid communities stand to benefit from the provision of consistent and agreed web service interfaces for data resources and the systems that manage them.

This document presents a specification for a collection of *querying interfaces* for RDF(S) data resources, which extends interfaces defined in the Web Services Data Access and Integration document [WS-DAI]. It also presents interfaces for handling RDF graphs in RDF(S) data resources. This specification can be applied in regular web services environments or as part of a grid fabric.

Contents

1 Introduction

RDF data access plays a central role in many types of semantic grid applications. By data access we mean the ability to retrieve, manipulate or insert data into an RDF(S) data resource.

This document presents a specification for a collection of *querying interfaces* for RDF(S) data resources. It also presents interfaces for handling RDF graphs in RDF(S) data resources. An RDF(S) data resource is a data source/sink that is based on the RDF data model, together with any associated management infrastructure that exhibits capabilities that are characteristics of RDF repositories. The management infrastructure may also exhibit RDF(S) model based views, exposing RDF Schema entailment capabilities over the resource.

This document should be read in conjunction with the generic *Web Services Data Access and Integration* specification [WS-DAI], which defines base interfaces that are extended in this document to cater for RDF(S) data resources. Also, this document should be read in conjunction with the WS-DAI RDF(S) informational document [DAIRDFS], which motivates this specification and outlines its relationship with other WS-DAI specifications. These specifications have been developed for representing data resources as web services, and form part of a broader activity within the Open Grid Forum to develop the Open Grid Services Architecture (OGSA) [OGSA].

1.1 Specification Scope

The base interfaces and properties for data access services are described in the *Web Services Data Access and Integration* specification [WS-DAI]. This specification extends these interfaces to allow access to and provide descriptions of RDF(S) data resources. RDF(S) data resources are assumed to contain a collection of RDF graphs which are accessed using the SPARQL [SPARQL] query language.

1.2 Specification Organization

The specification is described using the notational conventions and terminology defined in sections 2 and 3. Sections 4, 5 and 6 present three RDF(S) interfaces: Collection, SPARQL and SPARQLItemsSet respectively. A mapping of the functionality presented is made to WSDL in Section 7. Section 8 discusses security. Section 9 draws conclusions.

1.3 Interface Composition

This specification does not mandate how interfaces are composed into services; the proposed interfaces may be used in isolation or in conjunction with others. Viable compositions of interfaces will, initially, follow established patterns for data access.

2 Notational Conventions

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in RFC-2119 [RFC2119].

When describing concrete XML Schemas and XML instance fragments, this specification uses the notational convention of [WS-Security]. Specifically, each member of an element’s children or attributes property is described using an XPath-like notation (e.g., /x:MyHeader/x:SomeProperty/@value1 indicates that namespace *x* is being used, the root element *MyHeader* and a child element *SomeProperty* with an attribute *value1*). The use of {any} indicates the presence of an element wildcard (<xsd:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xsd:anyAttribute/>).

In the body of the specification, when patterns of messages are described, the layout of the XML of each message is presented, as opposed to the XML Schema; the XML Schema is provided in the appendices. The following notation is used to indicate the cardinality of XML elements in XML fragments:

* zero or more
+ one or more
? zero or one

Where no notation is added to an element, one instance of the element is expected.

This specification generally adopts the terminology defined in the Open Grid Services Architecture Glossary of Terms [OGSA Glossary] and W3C Glossary [W3C Glossary]. This terminology is extended in Section 3.

This specification uses namespace prefixes throughout; these are listed in the table below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Prefix	Namespace
wsa	http://www.w3.org/2005/08/addressing
wsdai	http://www.ggf.org/namespaces/2005/12/WS-DAI
wsdairdfs	http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query
xsd	http://www.w3.org/2001/XMLSchema

3 Terminology

Model-independent terminology, i.e., data resource, data access service, consumer and data set, is given in the WS-DAI specification [WS-DAI].

3.1 RDF(S) Data Resource

An RDF(S) Data Resource is a data source or sink that is based on the RDF data model, together with any associated management infrastructure that exhibits capabilities that are characteristic of RDF repositories. The management infrastructure may also exhibit RDF(S) model based views, exposing RDF Schema entailment capabilities over the resource. The following RDF(S) Data Resources are defined in this document:

- RDF(S) Collection: provides management-based operations to a collection of RDF graphs.
- SPARQL: provides query-based access to a collection of RDF graphs.
- SPARQLItemsSet: a resource used to access the results of SPARQL queries.

3.2 RDF(S) Interfaces

The word interface refers to the collections of messages and XML structures that describe the ways in which a consumer can validly interact, through this and the WS-DAI specifications, with a data access service. It is not intended to refer specifically to the proposed use of the word interface found in WSDL 2.0 [WSDL-2], although this may be an appropriate mapping in the future.

This specification extends the base interfaces and corresponding properties defined in the WS-DAI specification to provide access to RDF(S) data resources. These data resources are considered to consist of a collection of RDF graphs. To cater for this representation, the following description and direct access interface are defined:

- RDFSCollectionDescription: exposes various properties of an RDF(S) Collection that a data access service may represent.
- RDFSCollectionAccess: provides access to RDF graphs in a collection.

Direct access to a data access service allows the results of a request to be delivered to the consumer directly in the response message. To cater for this mode of operation the following interface is defined for accessing an RDF(S) data resource using established query language:

- SPARQLAccess: allows the evaluation of SPARQL statements across a collection of RDF graphs.

Indirect access is supported through the use of the factory pattern. This allows data, usually the result of a query, to be accessed by way of a new service-managed data resource, and thus the data is not returned directly to the consumer. To cater for this mode of operation the following interface is defined:

- SPARQLFactory: provides access to the results of a SPARQL query.

To support access to the data resources resulting from the use of the factory pattern, additional description and direct access interfaces are defined, in particular:

- SPARQLItemsSetDescription: provides properties describing a set of SPARQL query result items.
- SPARQLResultsSetAccess: provides access to a set of query results, which are the result of a SPARQL select/ask query.
- SPARQLTriplesSetAccess: provides access to a set of triples, which are the result of a SPARQL construct/describe query.

These interfaces, and their corresponding properties, are specified in sections 4 to 6.

3.3 Relationships with other specifications

The WS-DAI-RDF(S) Querying Specification does not define its own query/update languages for RDF(S) data resources. Instead, it acts as a channel for RDF query and update languages to be conveyed to the appropriate data resources, for instance RDF(S) data resources or a data resource that supports RDF type queries. In this document, interface support is provided for a language based on the following standard:

- SPARQL Query Language for RDF [SPARQL]: a query language for RDF data.

Some messages and XML structures are defined based on the following standards:

- SPARQL Query Results XML Format [RESULTS]: is an XML format for the variable bindings and boolean result format provided by the SPARQL query language.
- SPARQL Protocol for RDF [SPROT]: is a communication protocol that allows clients to execute queries over the Internet using SPARQL query processors.

These three specifications are W3C recommendations. The patterns defined in this document can also be employed to encompass new or emerging SPARQL versions, e.g. [SPARQL11].

4 RDF(S) Collection

4.1 Static RDF(S) Collection Description

The elements described in this section extend those defined in the WS-DAI specification. They are contained within an RDFSCollectionPropertyDocument element that extends the PropertyDocument type defined in the WS-DAI specification. In this specification, which models RDF from a querying viewpoint, an RDF(S) collection is composed of a set of RDF graphs.

4.1.1 RDFSCollection

```
 <xsd:complexType name="GraphDescriptionType">
  <xsd:attribute name="name" type="xsd:anyURI"/>
</xsd:complexType>

<xsd:complexType name="CollectionType">
  <xsd:sequence>
    <xsd:element name="Graph" type="wsdairdfs:GraphDescriptionType"
                 minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:element name="RDFSCollection" type="wsdairdfs:CollectionType"/>
```

/wsdairdfs:RDFSCollectionPropertyDocument/wsdairdfs:RDFSCollection/wsdairdfs:Graph
The string name that uniquely identifies an RDF graph within the RDF(S) collection.

4.1.2 NumberOfGraphs

```
<xsd:element name="NumberOfGraphs" type="xsd:unsignedLong"/>
```

/wsdairdfs:RDFSPropertyDocument/wsdairdfs:NumberOfGraphs
The number of graphs in the data resource.

4.2 Configurable RDF(S) Collection Description

No additional configurable properties are defined.

4.3 Example of RDFSCollectionPropertyDocument

A non-normative example of a RDFSCollection PropertyDocument follows:

```
<wsdairdfs:RDFSCollectionPropertyDocument xmlns:...>
  <wsdai:DataResourceAbstractName>
     urn:dais:ds1
  </wsdai:DataResourceAbstractName>
  <wsdai:DataResourceManagement>ExternallyManaged</wsdai:DataResourceManagement>
  <wsdai:DatasetMap>
    <wsdai:MessageQName>wsdairdfs:AddGraphs</wsdai:MessageQName>
    <wsdai:DatasetFormatURI>
      http://www.ogf.org/schemas/somedocumentschema
    </wsdai:DatasetFormatURI>
  </wsdai:DatasetMap>
  <wsdai:ConfigurationMap>
    <wsdai:MessageQName>wsdairdfs:GraphSelectionFactory</wsdai:MessageQName>
    <wsdai:PortTypeQName>wsdairdfs:RDFSCollectionAccess</wsdai:PortTypeQName>
    <wsdai:ConfigurationDocumentQName>
         wsdairdfs:RDFSCollectionConfigurationDocument
    </wsdai:ConfigurationDocumentQName>
    <DefaultConfigurationDocument>
      <wsdai:ConfigurationDocument>
        <wsdai:DataResourceDescription/>
        <wsdai:Readable>true</wsdai:Readable>
        <wsdai:Writeable>true</wsdai:Writeable>
        <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
        <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
        <wsdai:ChildSensitiveToParent>Insensitive</wsdai:ChildSensitiveToParent>
        <wsdai:ParentSensitiveToChild>Insensitive</wsdai:ParentSensitiveToChild>
      </wsdai:ConfigurationDocument>
    </DefaultConfigurationDocument>
  </wsdai:ConfigurationMap>
  <wsdai:DataResourceDescription/>
  <wsdai:Readable>true</wsdai:Readable>
  <wsdai:Writeable>true</wsdai:Writeable>
  <wsdai:ConcurrentAccess>true</wsdai:ConcurrentAccess>
  <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
  <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
  <wsdai:ChildSensitiveToParent>Insensitive</wsdai:ChildSensitiveToParent>
  <wsdai:ParentSensitiveToChild>Insensitive</wsdai:ParentSensitiveToChild>

  <wsdairdfs:RDFSCollection name="urn:dais">
    <!—RDF graphs belonging to the RDFSCollection ->
    <Graph name="http://example.org/graph1"/>
    <Graph name="http://example.org/graph2"/>
    <Graph name="http://example.org/graph3"/>
  </wsdairdfs:RDFSCollection>
```

```
    <wsdairdfs:NumberOfGraphs>3</wsdairdfs:NumberOfGraphs>

</wsdairdfs:RDFSCollectionPropertyDocument>
```

4.4 RDFSCollectionAccess

The *RDFSCollectionAccess* interface provides access to a collection of RDF graphs. AddGraphs and RemoveGraphs should be supported only when SPARQL 1.0 [SPARQL] is used.

4.4.1 RDFSCollectionAccess::GetCollectionPropertyDocument

Allows a copy of the CollectionPropertyDocument document to be retrieved.

Input

- GetCollectionPropertyDocumentRequest
    - DataResourceAbstractName – the abstract name of the resource from which the properties are to be obtained.

Output

- GetCollectionPropertyDocumentResponse
    - PropertyDocument – the properties described in the data description section.

Faults

- InvalidResourceNameFault – the supplied data resource abstract name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.

4.4.2 RDFSCollectionAccess::AddGraphs

Optional operation for resources that do not support updates via SPARQL. Create new RDF graphs in the specified collection. An attempt should be made to add all the graphs even if some additions fail. When a graph with the same name already exists, it will be overwritten.

Input

- AddGraphsRequest
    - DataResourceAbstractName –the abstract name of the data resource to which the message is directed.
    - AddGraphRequestWrapper+ – for each graph:
        - GraphNameURI – the URI of the new graph - used to identify the graph within this RDF(S) collection.
        - Data – the content of the graph, serialized using RDF/XML [RDF-SYNTAX].

Output

- AddGraphsResponse
    - AddGraphsResponseWrapper+ – a wrapper element for each graph:
        - GraphNameURI – the URI of the graph.
        - Response – result of the add operation. Possible values for this are:
            - Success: the graph was successfully added.
            - GraphNotAdded-NotAuthorized: Graph not added as client is not authorized.
            - GraphOfSameNameOverwritten - existing graph of same name is overwritten.
            - Failure - any other error.

        - Detail – Details on the add operation such as warnings and error messages in the case of a failure to add the graph.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – The service is already processing a request and ConcurrentAccess is false.

4.4.3 RDFSCollectionAccess::GetGraphs

Retrieve RDF graphs from the specified collection. An attempt should be made to retrieve all the graphs even if some graphs do not exist.

Input

- GetGraphsRequest
    - DataResourceAbstractName –the abstract name of the data resource to which the message is directed.
    - GetGraphsRequestWrapper+ – for each graph:
        - GraphNameURI – the URI of the graph to be retrieved.

Output

- GetGraphsResponse
    - GetGraphsResponseWrapper+ – for each graphs:
        - GraphNameURI – the URI of a graph.
        - Response – result of the retrieval operation. Possible values for this are:
            - Success.
            - Graph not retrieved as it does not exist.
        - Data – the content of the graph retrieved, serialized using RDF/XML
        - Detail – Details on the get operation such as warnings and error message.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – The service is already processing a request and ConcurrentAccess is false.
- DatasetTooLargeFault - the total size of the data produced by the request was too large to be sent back to the client. The client should reduce the number of requested graphs or attempt to use indirect access.

4.4.4 RDFSCollectionAccess::RemoveGraphs

Optional operation for resources that do not support updates via SPARQL. Remove a set of graphs from the collection. An attempt should be made to remove all of the graphs even if some removals fail.

Input

- RemoveGraphsRequest
    - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
    - RemoveGraphRequestWrapper+ - for each graphs:
        - GraphNameURI – the URI of each graph to be removed.

Output

- RemoveGraphsResponse

- RemoveGraphsResponseWrapper+ – for each graph, the input graph name and the result of the remove operation is recorded.
  - GraphNameURI – the URI of the graph response applies to.
  - Response – result of the remove operation. Possible value for this are,:
    - Success.
    - GraphNotRemoved-NotAuthorized - graph not removed as the client is not authorized.
    - GraphNotRemoved-GraphDoesNotExist - graph not removed as the graph name URI specified does not exist in the collection.
  - Detail – Details on the remove operation such as warnings and error messages.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.

4.5 RDFSCollectionFactory

4.5.1 RDFSCollectionFactory::GraphSelectionFactory

Returns an endpoint reference to a data resource that represents an existing RDF graph in the RDF(S) collection. This factory operation could be used if access to an individual RDF graph through a different port type is required.

Input

- GraphSelectionFactoryRequest
  - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
  - PortTypeQName? – the QName of the portType through which the resulting data should be accessed. The QName value here MUST correspond to one that is specified in the ConfigurationMap property.
  - ConfigurationDocument? – a document that specifies the properties of the data resource that is to be used to access the data.
  - PreferredTargetService? – the EPR of the preferred service that is to act as the host for the new data resource.
  - GraphNameURI – the URI of the graph to be exposed through a service.

Output

- GraphSelectionFactoryResponse
  - DataResourceAddress – a data resource address.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.
- GraphDoesNotExistFault – the specified graph could not be found.
- InvalidPortTypeQNameFault – the PortTypeQNname specified is not in the collection defined by ConfigurationMap property.
- InvalidConfigurationDocumentFault – the ConfigurationDocument specified is not valid according to the ConfigurationDocumentQName when the ConfigurationMap is indexed by the specified PortTypeQName.

5 SPARQL

The element described in this section extend those defined in the WS-DAI specification. It is contained within a SPARQLPropertyDocument element that extends the PropertyDocument type defined in the WS-DAI specification.

5.1 Static SPARQL Description

```
<xsd:element name="ExternalGraphAccess" type="xsd:boolean"/>
```

/wsdairdfs:SPARQLPropertyDocument/wsdairdfs:ExternalGraphAccess

Indicates whether the data access service supports accessing external RDF graphs, i.e. RDF graphs that are not managed by the connected RDF(S) data resource.

A SPARQL query may specify a collection of RDF graphs (called *RDF Dataset* [SPARQL]) against which it will be executed. This element informs the consumer whether external RDF graphs are allowed to be included in the RDF Dataset.

5.2 Configurable SPARQL Description

No extra configurable properties are defined for SPARQL.

5.3 DatasetMap

The DatasetMap element refers to the format of the returned dataset. Services implementing the SPARQLAccess interface MUST support at least the W3C SPARQL Query Result XML format [RESULTS] and RDF/XML serialization. The former is for the results returned by a SPARQL select/ask query while the latter is for the results returned by a SPARQL construct/describe query. The services may optionally support the RDF/N3 format [N3-Primer] for the SPARQL construct/describe query.

5.4 Example of SPARQLPropertyDocument

A non-normative example of a SPARQLPropertyDocument follows:

```
<wsdai:SPARQLPropertyDocument xmlns:...>
  <wsdai:DataResourceAbstractName>urn:dais:ds2
  </wsdai:DataResourceAbstractName>
  <wsdai:DataResourceManagement>
     ExternallyManaged
  </wsdai:DataResourceManagement>
  <wsdai:ParentDataResource>
    <wsa:Address>http://www.ogf.org/services/daisservice</wsa:Address>
    <wsa:ReferenceParameters>
      <DataResourceAbstractName>urn:dais:ds1</DataResourceAbstractName>
    </wsa:ReferenceParameters>
    <wsa:Metadata/>
  </wsdai:ParentDataResource>

  <wsdai:DatasetMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:DatasetFormatURI>
       http://www.w3.org/2005/sparql-results#sparql
    </wsdai:DatasetFormatURI>
  </wsdai:DatasetMap>
  <wsdai:DatasetMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:DatasetFormatURI>http://www.w3.org/1999/02/22-rdf-syntax-ns
    </wsdai:DatasetFormatURI>
  </wsdai:DatasetMap>
  <wsdai:DatasetMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:DatasetFormatURI>rdf/n3</wsdai:DatasetFormatURI>
  </wsdai:DatasetMap>
```

```
  <wsdai:ConfigurationMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecuteFactory</wsdai:MessageQName>
<wsdai:PortTypeQName>wsdairdfs:SPARQLQueryResultsAccessPT</wsdai:PortTypeQName>
    <wsdai:ConfigurationDocumentQName>
       wsdairdfs:SPARQLQueryResultsConfigurationDocument
    </wsdai:ConfigurationDocumentQName>
    <DefaultConfigurationDocument>
      <wsdai:ConfigurationDocument>
        <wsdai:DataResourceDescription/>
        <wsdai:Readable>true</wsdai:Readable>
        <wsdai:Writeable>false</wsdai:Writeable>
        <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
        <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
        <wsdai:ChildSensitiveToParent>Insensitive</wsdai:ChildSensitiveToParent>
        <wsdai:ParentSensitiveToChild>Insensitive</wsdai:ParentSensitiveToChild>
      </wsdai:ConfigurationDocument>
    </DefaultConfigurationDocument>
  </wsdai:ConfigurationMap>

  <wsdai:LanguageMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:LanguageURI>
              http://www.w3.org/TR/rdf-sparql-query/
    </wsdai:LanguageURI>
  </wsdai:LanguageMap>
  <wsdai:DataResourceDescription/>
  <wsdai:Readable>true</wsdai:Readable>
  <wsdai:Writeable>false</wsdai:Writeable>
  <wsdai:ConcurrentAccess>true</wsdai:ConcurrentAccess>
  <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
  <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
  <wsdai:ChildSensitiveToParent>Insensitive</wsdai:ChildSensitiveToParent>
  <wsdai:ParentSensitiveToChild>Insensitive</wsdai:ParentSensitiveToChild>

  <wsdairdfs:ExternalGraphAccess>true</wsdairdfs:ExternalGraphAccess>
</wsdai:PropertyDocument>
```

5.5 SPARQLAccess

This interface supports SPARQL requests to be made to an RDF(S) data resource. An RDF(S) data access service MUST implement the SPARQL operations and expose the SPARQL Description properties. In this example a consumer uses the SPARQLExecute message to submit a RequestDocument in a format defined in [SPROT]. The associated SPARQLQueryExecuteResponse message will contain a set of query result items.

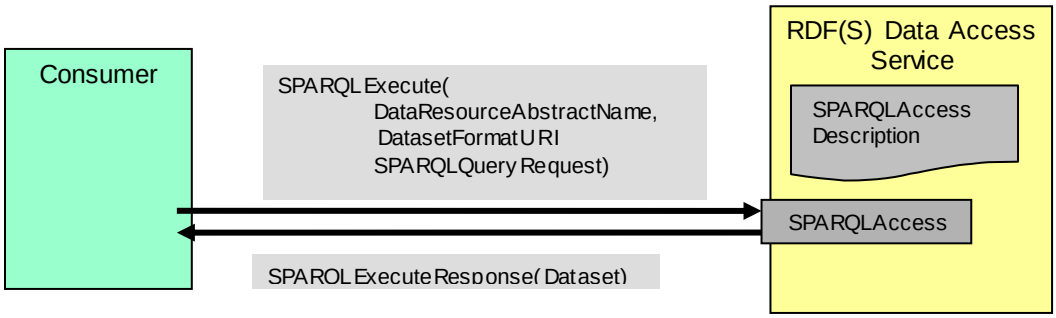

Figure 1: Overview – SPARQLAccess

5.5.1 SPARQLAccess::GetSPARQLPropertyDocument

Allows a copy of the SPARQLPropertyDocument document to be retrieved.

Input

- GetSPARQLPropertyDocumentRequest
    - DataResourceAbstractName – the abstract name of the resource from which the properties are to be obtained.

Output

- GetSPARQLPropertyDocumentResponse
    - PropertyDocument – the properties described in the data description section.

Faults

- InvalidResourceNameFault – the supplied data resource abstract name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.

5.5.2 SPARQLAccess::SPARQLExecute

Execute a SPARQL query statement. SPARQL 1.0 is restricted to ASK, SELECT, CONSTRUCT and DESRCIBE query forms however, SPARQL 1.1 will permit updates [SPARQL-UPDATE] which may also be encompassed by this operation.

Input

- SPARQLExecuteRequest
    - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
    - DatasetFormatURI? – the format used to represent the SPARQL query result format in the corresponding SPARQLExecuteResponse. The format of the SPARQL select/ask query results are always encoded using W3C SPARQL Query Result XML format [RESULTS]. For SPARQL construct/describe query results, the default format is RDF/XML format [RDF-SYNTAX], but the consumer may specify other result formats such as RDF/N3 by using this parameter.
    - SPARQLQueryRequest – SPARQL Query request format including any SPARQL query string and a set of RDF graph URIs. The format is based on the input type of the *query* operation defined in [SPROT] which is shown below.

```
<xsd:element name="query-request">
 <xsd:complexType>
  <xsd:sequence>
     <xsd:element minOccurs="1" maxOccurs="1" name="query" type="xs:string">
          <xsd:annotation>
          <xsd:documentation>query is an xsd:string constrained by the language definition,
           http://www.w3.org/TR/rdf-sparql-query/#grammar, as "a sequence of characters in the
           language defined by the [SPARQL] grammar, starting with the Query
           production"</xsd:documentation>
          <xsd:annotation>
         </xsd:element>
<xsd:element minOccurs="0" maxOccurs="unbounded" name="default-graph-uri" type="xsd:anyURI" />
<xsd:element minOccurs="0" maxOccurs="unbounded" name="named-graph-uri" type="xsd:anyURI" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Output

- SPARQLExecuteResponse
    - Dataset? - optional because updates MAY return an empty response message that indicates that the result of the update (or sequence of updates) was successful. Updates MAY also specify some details about a successful update operation using an appropriate dataset.

- DatasetFormatURI – specifies how the result items set is serialized. The value MUST correspond to one specified in DatasetMap at the data access service (see the WS-DAI specification for details).
- DatasetData – the results from the request, a set of query result items.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- InvalidDatasetFormatFault – the supplied dataset format is not known to the service.
- InvalidLanguageFault – the supplied expression language is not known to the service.
- InvalidExpressionFault – the supplied expression is not of a form known to the service.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.
- SPARQLFault – the requested SPARQL operation failed during execution.
  - Detail? – Describe details of the SPARQL operation failure.
- ExternalGraphFault – the service is not able to process external graphs specified in the query request.
- DatasetTooLargeFault – the total size of the data produced by the request was too large to be sent back to the client. The client should attempt to use indirect access.

5.6 SPARQLFactory

The example in Figure 2 presents a SPARQLFactory interface. The SPARQLExecuteFactory operation is used to make the results of a query available through, potentially, a separate data access service; for example, a data access service which implements the SPARQLItemsSet port type. In this example the SPARQLItemsSet could be stored in a database or decoupled from the database, but the important distinction is that the data is represented as a set of query result items that does not implement the SPARQLAccess portType and hence does not provide facilities for submitting SPARQL expressions.

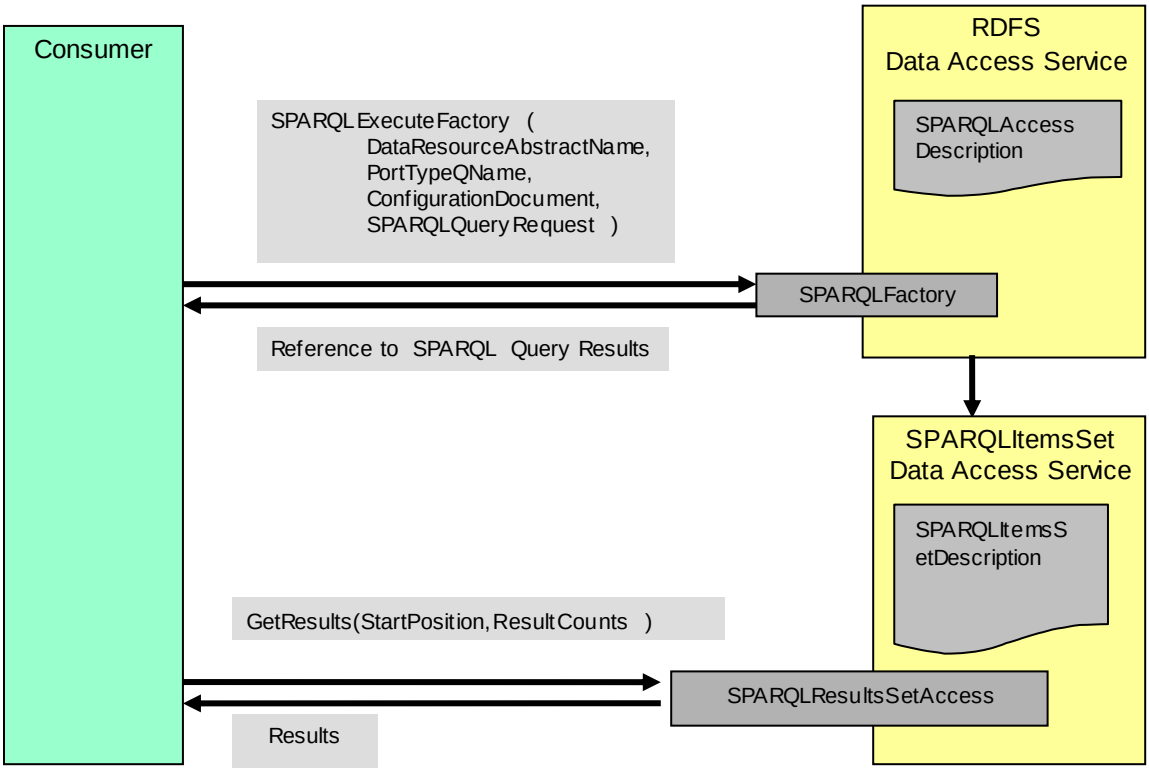

Figure 2: Overview – SPARQLFactory

5.6.1 SPARQLFactory::SPARQLExecuteFactory

Provide access to the results of a SPARQL request in a separate data access service. This separate data access service SHOULD be a service that implements the SPARQLResultsSetAccess or SPARQLTriplesSetAccess interface.

Input

- SPARQLExecuteFactoryRequest
  - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
  - PortTypeQName? – the QName of the portType through which the resulting data should be accessed. The QName value here MUST correspond to one that is specified in the ConfigurationMap property.
  - ConfigurationDocument? – a document that specifies the properties of the data resource that is to be used to access the data.
  - PreferredTargetService? – the EPR of the preferred service that is to act as the host for the new data resource.
  - SPARQLQueryRequest – SPARQL Query request format including any SPARQL query string and the set of RDF graph URIs. The format is based on the input type of the *query* operation defined in [SPROT].

Output

- SPARQLExecuteFactoryResponse
  - DataResourceAddress – a data resource address.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- InvalidPortTypeQNameFault – the PortTypeQNname specified is not in the collection defined by ConfigurationMap property.
- InvalidConfigurationDocumentFault – the ConfigurationDocument specified is not valid according to the ConfigurationDocumentQName when the ConfigurationMap is indexed by the specified PortTypeQName.
- InvalidLanguageFault – the supplied expression language is not known to the service.
- InvalidExpressionFault – the supplied expression is not of a form known to the service.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.
- SPARQLFault – the requested SPARQL operation failed during execution.
  - Detail? – Describe details of the SPARQL operation failure.

6 SPARQLItemsSet

6.1 Static SPARQLItemsSet Description

6.1.1 NumberOfItems

```
<xsd:element name="NumberOfItems" type="xsd:unsignedLong"/>
```

/wsdairdfs:SPARQLItemsSetPropertyDocument/wsdairdfs:NumberOfItems
The total number of items in the query result.

6.2 Configurable SPARQLItemsSet Description

No extra configurable properties are defined for SPARQLItemsSet.

6.3 DatasetMap

The DatasetMap element refers to the format of the returned dataset. Services implementing SPARQLResultsSetAccess interface must support the W3C SPARQL Query Result XML format [RESULTS] while those implementing SPARQLTriplesSetAccess interface must support the RDF/XML format [RDF-SYNTAX]. Since each query type has a different result format, the consumer needs to know from which query type the results come from before retrieving them. This can be done by inspecting this property's value.

6.4 Example SPARQLItemsSetPropertyDocument

```
<wsdairdfs:SPARQLItemsSetPropertyDocument xmlns:...>
  <wsdai:DataResourceAbstractName>
     urn:dais:ds2
  </wsdai:DataResourceAbstractName>
  <wsdai:DataResourceManagement>ExternallyManage
  </wsdai:DataResourceManagement>
  <wsdai:ParentDataResource>
    <wsa:Address>http://www.ggf.org/services/daisservice</wsa:Address>
    <wsa:ReferenceParameters>
      <DataResourceAbstractName>urn:dais:ds1</DataResourceAbstractName>
    </wsa:ReferenceParameters>
    <wsa:Metadata/>
  </wsdai:ParentDataResource>
  <wsdai:DatasetMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:DatasetFormatURI>
       http://www.w3.org/2005/sparql-results#sparql
    </wsdai:DatasetFormatURI>
  </wsdai:DatasetMap>

  <wsdai:ConfigurationMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecuteFactory</wsdai:MessageQName>
<wsdai:PortTypeQName>wsdairdfs:SPARQLItemsSetAccessPT</wsdai:PortTypeQName>
<wsdai:ConfigurationDocumentQName>wsdairdfs:SPARQLItemsSetConfigurationDocument
    </wsdai:ConfigurationDocumentQName>
    <DefaultConfigurationDocument>
      <wsdai:ConfigurationDocument>
        <wsdai:DataResourceDescription/>
        <wsdai:Readable>true</wsdai:Readable>
        <wsdai:Writeable>false</wsdai:Writeable>
        <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
        <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
        <wsdai:ChildSensitiveToParent>Sensitive</wsdai:ChildSensitiveToParent>
        <wsdai:ParentSensitiveToChild>Sensitive</wsdai:ParentSensitiveToChild>
      </wsdai:ConfigurationDocument>
    </DefaultConfigurationDocument>

  <wsdai:LanguageMap>
    <wsdai:MessageQName>wsdairdfs:SPARQLExecute</wsdai:MessageQName>
    <wsdai:LanguageURI>
       http://www.w3.org/TR/rdf-sparql-query/
    </wsdai:LanguageURI>
  </wsdai:LanguageMap>

  </wsdai:ConfigurationMap>
  <wsdai:DataResourceDescription/>
  <wsdai:Readable>true</wsdai:Readable>
  <wsdai:Writeable>false</wsdai:Writeable>
  <wsdai:ConcurrentAccess>true</wsdai:ConcurrentAccess>
  <wsdai:TransactionInitiation>NotSupported</wsdai:TransactionInitiation>
  <wsdai:TransactionIsolation>NotSupported</wsdai:TransactionIsolation>
  <wsdai:ChildSensitiveToParent>Sensitive</wsdai:ChildSensitiveToParent>
  <wsdai:ParentSensitiveToChild>Sensitive</wsdai:ParentSensitiveToChild>
```

```
  <wsdairdfs:NumberOfItems>100</wsdairdfs:NumberOfItems>
</wsdairdfs:SPARQLItemsSetPropertyDocument>
```

6.5 SPARQLResultsSetAccess

This interface provides access to query results returned by a SPARQL select/ask query. An RDF data access service may implement the SPARQLResultsSetAccess interface and expose the SPARQLItemsSetDescription properties. A consumer uses the GetResults message to retrieve a number of results from the items set. It submits a RequestData containing the StartPosition and ResultCount parameters. The associated GetResultsResponse message will contain the requested results in a serialized form.

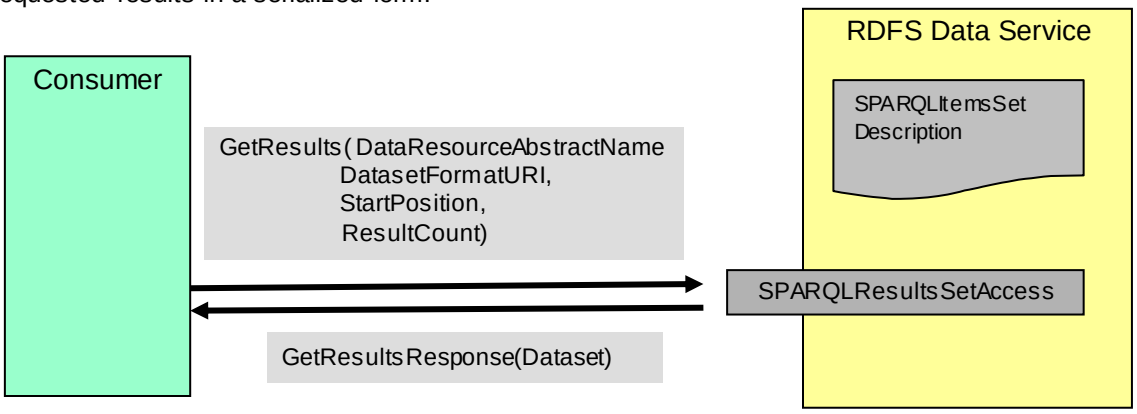

Figure 3: SPARQLResultsSetAccess

6.5.1 SPARQLResultsSetAccess::GetSPARQLItemsSetPropertyDocument

Allows a copy of the SPARQLItemsSetsPropertyDocument document to be retrieved.

Input

- GetSPARQLItemsSetPropertyDocumentRequest
  - DataResourceAbstractName – the abstract name of the resource.

Output

- GetSPARQLItemsSetPropertyDocumentResponse
  - PropertyDocument – the properties described in the data description section.

Faults

- InvalidResourceNameFault – the supplied data resource abstract name is not known to the service.

6.5.2 SPARQLResultsSetAccess::GetResults

Returns a specified number of results obtained from a SPARQL SELECT/ASK query, i.e. a sequence of variable bindings or boolean values.

Input

- GetResultsRequest
  - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
  - DatasetFormatURI? – the URI of the format of the return result format.
  - StartPosition – the position of the first result to be returned. (Sequence starts with position 0).
  - ResultCount – the number of results to be returned.

Output

- GetResultsResponse
  - Dataset

- DatasetFormatURI – the format of the data being used to return the results.
- DatasetData – the results from the request, a set of query result items.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- InvalidDatasetFormatFault – the supplied dataset format is not known to the service.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.
- InvalidStartPositionFault – The start position is not valid.
- InvalidCountFault – Cannot return this number of results.
- DatasetTooLargeFault - the total size of the data produced by the request was too large to be sent back to the client. The client should attempt to use indirect access.

6.6 SPARQLTriplesSetAccess

This interface provides access to query results returned by a SPARQL construct/describe query. An RDF data access service may implement the SPARQLTriplesSetAccess interface and expose the SPARQLItemsSetDescription properties. A consumer uses the GetTriples message to retrieve a number of triples from the items set (RDF graph). It submits a RequestData containing the StartPosition and ResultCount parameters. The associated GetTriplesResponse message will contain the requested triples in a serialized form.

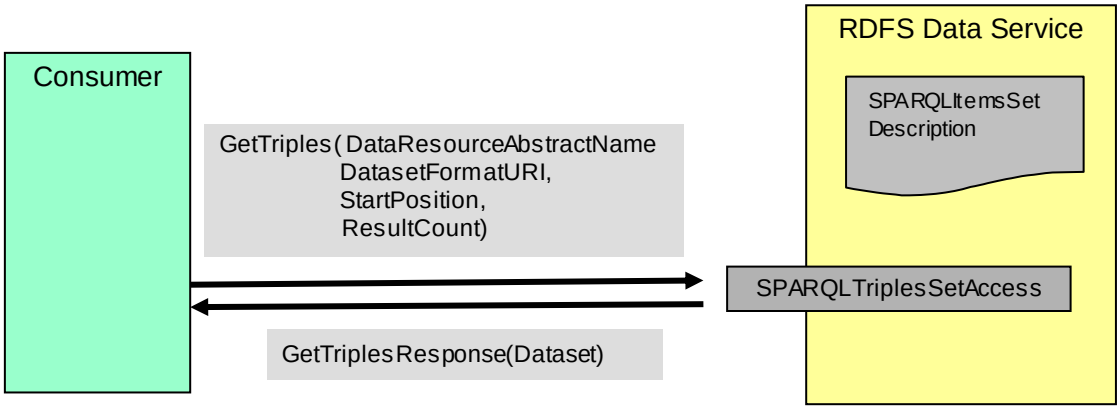

Figure 4: SPARQLTriplesSetAccess

6.6.1 SPARQLTriplesSetAccess::GetSPARQLItemsSetPropertyDocument

Allows a copy of the SPARQLItemsSetPropertyDocument to be retrieved.

Input

- GetSPARQLItemsSetPropertyDocumentRequest
  - DataResourceAbstractName – the abstract name of the data resource.

Output

- GetSPARQLItemsSetPropertyDocumentResponse
  - PropertyDocument – the properties described in the data description section.

Faults

- InvalidResourceNameFault – the supplied data resource abstract name is not known to the service.

6.6.2 SPARQLTriplesSetAccess::GetTriples

Returns a specified number of triples.

Input

- GetTriplesRequest
  - DataResourceAbstractName – the abstract name of the data resource to which the message is directed.
  - DatasetFormatURI? – the URI of the format of the return result format.
  - StartPosition – the position of the first triple to be returned. (Sequence starts with position 0).
  - ResultCount – the number of triples to be returned.

Output

- GetTriplesResponse
  - Dataset
    - DatasetFormatURI – the format of the data being used to return the triples.
    - DatasetData – the results from the request, a set of query result items.

Faults

- InvalidResourceNameFault – the supplied resource name is not known to the service.
- DataResourceUnavailableFault – the specified data resource is unavailable.
- InvalidDatasetFormatFault – the supplied dataset format is not known to the service.
- NotAuthorizedFault – the consumer is not authorized to perform this operation at this time.
- ServiceBusyFault – the service is already processing a request and ConcurrentAccess is false.
- InvalidStartPositionFault – The start position is not valid.
- InvalidCountFault – Cannot return this number of results.
- DatasetTooLargeFault - the total size of the data produced by the request was too large to be sent back to the client. The client should reduce the number of requested graphs or attempt to use indirect access.

7 Mapping to WSDL

For a mapping to WSDL see the following sections:

- RDFS Collection
  - XML Schema – Appendix A.1.
  - WSDL – Appendix A.2.
- SPARQLAccess
  - XML Schema – Appendix B.1.
  - WSDL – Appendix B.2.
- SPARQLItemsSet
  - XML Schema – Appendix C.1.
- SPARQLResultsSetAccess
  - WSDL – Appendix C.2.
- SPARQLTriplesSetAccess
  - WSDL – Appendix C.3.

8 Security Considerations

The realizations of a grid data access service will use standard web service security mechanisms as specified by other standards bodies. The assumption is that these standards will also indicate how to make information related to authentication, authorization security, etc., available.

.

9 Conclusion

This specification has presented a specialization of the interfaces defined in the *WS Data Access and Integration* [WS-DAI] specification providing the SPARQL querying capabilities required to address RDF(S) based data resources.

10 Author Information

Isao Kojima
Information Technology Research Institute
AIST
Umezono1-1-1
Tsukuba, Ibaraki, 305-8568
Japan

Said Mirza Pahlevi
AIST
Umezono 1-1-1
Tsukuba, Ibaraki, 305-8568
Japan

Steven Lynden
Information Technology Research Institute
AIST
Umezono1-1-1
Tsukuba, Ibaraki, 305-8568
Japan

11 Contributors

Carlos Buil Aranda, UPM
Oscar Corcho, UPM
Miguel Esteban Gutiérrez, UPM
Masahiro Kimoto, AIST
Akiyoshi Matono, AIST

12 Acknowledgements

The Database Access and Integration Services (DAIS) Working Group of the Open Grid Forum has been active over several years, and many people have contributed to discussions within the group, including but not limited to: Mario Antonioletti, Amy Krause, Norman Paton, Dave Pearson, and Asunción Gómez-Pérez

13 Intellectual Property Statement

14 Full Copyright Notice

15 References

[OGSA]
I.Foster (Ed), H. Kishimoto (Ed), A. Savva (Ed), D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, R. Subramaniam, J. Treadwell, J. Von Reich. The Open Grid Services Architecture, Version 1.0. Global Grid Forum. GFD-I.030. 29 January 2005. http://www.ggf.org/documents/GFD.30.pdf.

[RFC2119]
S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, Internet Engineering Task Force, RFC 2119, http://www.ietf.org/rfc/rfc2119.txt, March 1997.

[OGSA Glossary]
J. Treadwell, *Open Grid Services Architecture Glossary of Terms*, GFD-I.044, January 25th 2005. http://www.ggf.org/documents/GFD.44.pdf.

[W3C Glossary]
http://www.w3.org/2003/glossary/

[WS-DAI]
M. Antonioletti, M. Atkinson, A. Krause, S. Malaika, S. Laws, N. W. Paton D. Pearson, and G. Riccardi. *Web Services Data Access and Integration – The Core (WS-DAI) Specification, Version 1.0*. GWD-R, Global Grid Forum, DAIS Working Group, Jun 2006.

[DAIRDFS]
I.Kojima, M.E.Guiterrez,O.Corcho,S.M.Pahlevi and A.G.Perez *DAIS for RDF(S) Realization – Introduction, Motivational Use Cases and Terminologies GFD-I. 163,December ,2009*. http://www.ogf.org/documents/GFD.163.pdf

[RDF]
Resource Description Framework (RDF)
http://www.w3.org/RDF/

[RDFS]
Dan Brickley and R.V.Guha(Eds.), *RDF Vocabulary Description Language 1.0: RDF Schema.* W3C Recommendation 10 February 2004
http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[RDF Semantics]
P. Hayes (Ed). *RDF Semantics*. W3C Recommendation 10 February 2004
http://www.w3.org/TR/rdf-mt/

[RDF-SYNTAX]
D.Beckett(Ed). *RDF/XML Syntax Specification*. W3C Recommendation 10 February 2004

[SPARQL]
Eric Prud'hommeaux and Andy Seaborne(Eds.). *SPARQL Query Language for RDF*, W3C Recommendation 15 January 2008.

http://www.w3.org/TR/rdf-sparql-query/

[SPARQL11]
Steve Harris and Andy Seaborne(Eds.) *SPARQL 1.1 Query language* ,W3C Working Draft 12 May 2011
http://www.w3.org/TR/sparql11-query/

[RESULTS]
D. Beckett and J.Broekstra(Eds.) *SPARQL Query Results XML Format*, W3C Recommendation 15 January 2008
http://www.w3.org/TR/rdf-sparql-XMLres/

[SPROT]
K. Clark, L. Feigenbaum and E.Torres(Eds.) *SPARQL Protocol for RDF*, W3C Recommendation 15 January 2008 http://www.w3.org/TR/rdf-sparql-protocol/

[N3-Primer]
Tim Berners-Lee, *Primer – Getting into RDF & Semantic Web using N3*, http://www.w3.org/2000/10/swap/Primer.html

[WSDL2]
Roberto Chinnici et al, *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*, http://www.w3.org/TR/wsdl20/

Appendix A.1 – RDFSCollection XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
        elementFormDefault="qualified"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
               xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

       <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI" schemaLocation="./wsdai core types.xsd"/>

    <!-- properties -->

       <!-- the descriptive properties for a RDFS graph  -->
       <xsd:complexType name="GraphDescriptionType">
              <xsd:attribute name="GraphnameURI" type="xsd:anyURI"/>
       </xsd:complexType>

       <!-- collection structure -->
       <xsd:complexType name="CollectionType">
              <xsd:sequence>
                     <xsd:element name="Graph" type="wsdairdfs:GraphDescriptionType" minOccurs="0" maxOccurs="unbounded"/>
              </xsd:sequence>
       </xsd:complexType>
       <xsd:element name="RDFSCollection" type="wsdairdfs:CollectionType"/>

       <!-- the number of Graphs in the data resource -->
       <xsd:element name="NumberOfGraphs" type="xsd:unsignedLong"/>

       <!-- property and configuration documents -->
       <xsd:complexType name="RDFSCollectionPropertyDocumentType">
              <xsd:complexContent>
                     <xsd:extension base="wsdai:PropertyDocumentType">
                            <xsd:sequence>
                                   <xsd:element ref="wsdairdfs:RDFSCollection"/>
                                   <xsd:element ref="wsdairdfs:NumberOfGraphs"/>
                            </xsd:sequence>
                     </xsd:extension>
              </xsd:complexContent>
       </xsd:complexType>
       <xsd:element name="RDFSCollectionPropertyDocument" type="wsdairdfs:RDFSCollectionPropertyDocumentType"/>
</xsd:schema>
```

Appendix A.2 – RDFSCollection WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdaidrdfs"
```

```
                              targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                              xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                              xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                        xmlns:wsa="http://www.w3.org/2005/08/addressing"
                              xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

<!-- WSDL IMPORTS ################################################## -->
       <wsdl:import location="./wsdai_core_porttypes.wsdl"
                            namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"/>
       <xsd:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="./ws-addressing-0805.xsd"/>

<!-- WSDL TYPES #################################################### -->
       <wsdl:types>
              <xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                                  elementFormDefault="qualified">
              <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                                  schemaLocation="./wsdai core types.xsd" />
              <xsd:include schemaLocation="./wsdairdfs_collection_types.xsd" />

              <!-- ################################## -->
              <!-- ###  GetGraphs Message Types   ### -->
              <!-- ################################## -->
                     <xsd:element name="GetGraphRequestWrapper">
                            <xsd:complexType>
                                   <xsd:sequence>
                                          <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                   </xsd:sequence>
                            </xsd:complexType>
                     </xsd:element>

                     <xsd:element name="GetGraphResponseWrapper">
                            <xsd:complexType>
                                   <xsd:sequence>
                                          <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                          <xsd:element name="Response">
                                                 <xsd:simpleType>
                                                        <xsd:restriction base="xsd:token">
                                                               <xsd:enumeration value="Success"/>
                                                               <xsd:enumeration value="GraphNotRetrieved-GraphDoesNotExist"/>
                                                        </xsd:restriction>
                                                 </xsd:simpleType>
                                          </xsd:element>
                                          <xsd:element name="Data" type="xsd:anyType" minOccurs="0"/>
                                          <xsd:element name="Detail" type="xsd:anyType" minOccurs="0"/>
                                   </xsd:sequence>
                            </xsd:complexType>
```

```
                </xsd:element>

 <xsd:element name="GetGraphsRequest">
   <xsd:complexType>
     <xsd:complexContent>
       <xsd:extension base="wsdai:BaseRequestType">
         <xsd:sequence>
           <xsd:element ref="wsdairdfs:GetGraphRequestWrapper"  minOccurs="1" maxOccurs="unbounded"/>
         </xsd:sequence>
       </xsd:extension>
     </xsd:complexContent>
       </xsd:complexType>
        </xsd:element>

<xsd:element name="GetGraphsResponse">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref="wsdairdfs:GetGraphResponseWrapper"  minOccurs="1" maxOccurs="unbounded"/>
                </xsd:sequence>
        </xsd:complexType>
</xsd:element>


<!-- ############################## -->
<!-- ### AddGraphs Message Types ### -->
<!-- ############################## -->
        <xsd:element name="AddGraphsRequestWrapper">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                <xsd:choice>
                                        <xsd:element name="Data" type="xsd:anyType"/>
                                </xsd:choice>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>

        <xsd:element name="AddGraphsResponseWrapper">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                <xsd:element name="Response">
                                        <xsd:simpleType>
                                                <xsd:restriction base="xsd:token">
                                                        <xsd:enumeration value="Success"/>
                                                        <xsd:enumeration value="GraphNotAdded-NotAuthorized"/>
                                                        <xsd:enumeration value="GraphOfSameNameOverwritten"/>
```

```
                                                        <xsd:enumeration value="Failure"/>
                                                </xsd:restriction>
                                        </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="Detail" type="xsd:anyType" minOccurs="0"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>

        <xsd:element name="AddGraphsRequest">
                <xsd:complexType>
                        <xsd:complexContent>
                                <xsd:extension base="wsdai:BaseRequestType">
                                        <xsd:sequence>
                                                <xsd:element ref="wsdairdfs:AddGraphsRequestWrapper"  minOccurs="1"
maxOccurs="unbounded"/>
                                        </xsd:sequence>
                                </xsd:extension>
                        </xsd:complexContent>
                </xsd:complexType>
        </xsd:element>

    <xsd:element name="AddGraphsResponse">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element ref="wsdairdfs:AddGraphsResponseWrapper"  minOccurs="1" maxOccurs="unbounded"/>
                </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <!-- ################################## -->
    <!-- ### RemoveGraphs Message Types ### -->
    <!-- ################################## -->

        <xsd:element name="RemoveGraphRequestWrapper">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>

        <xsd:element name="RemoveGraphResponseWrapper">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                        <xsd:element name="Response">
```

```
                                                <xsd:simpleType>
                                                       <xsd:restriction base="xsd:token">
                                                              <xsd:enumeration value="Success"/>
                                                              <xsd:enumeration value="GraphNotRemoved-NotAuthorized"/>
                                                              <xsd:enumeration value="GraphNotRemoved-GraphDoesNotExist"/>
                                                       </xsd:restriction>
                                                </xsd:simpleType>
                                         </xsd:element>
                                  <xsd:element name="Detail" type="xsd:anyType" minOccurs="0"/>
                           </xsd:sequence>
                    </xsd:complexType>
             </xsd:element>

             <xsd:element name="RemoveGraphsRequest">
                    <xsd:complexType>
                           <xsd:complexContent>
                                  <xsd:extension base="wsdai:BaseRequestType">
                                         <xsd:sequence>
                                                <xsd:element ref="wsdairdfs:RemoveGraphRequestWrapper"  minOccurs="1"
maxOccurs="unbounded"/>
                                         </xsd:sequence>
                                  </xsd:extension>
                           </xsd:complexContent>
                    </xsd:complexType>
      </xsd:element>

      <xsd:element name="RemoveGraphsResponse">
             <xsd:complexType>
                    <xsd:sequence>
                           <xsd:element ref="wsdairdfs:RemoveGraphResponseWrapper"  minOccurs="1" maxOccurs="unbounded"/>
                    </xsd:sequence>
             </xsd:complexType>
      </xsd:element>

      <!-- ################################### -->
      <!-- ###  GraphSelectionFactory Types ### -->
      <!-- ################################### -->

      <xsd:element name="GraphSelectionFactoryRequest">
             <xsd:complexType>
                    <xsd:complexContent>
                           <xsd:extension base="wsdai:FactoryRequestType">
                                  <xsd:sequence>
                                         <xsd:element name="GraphNameURI" type="xsd:anyURI"/>
                                  </xsd:sequence>
                           </xsd:extension>
                    </xsd:complexContent>
```

```
                </xsd:complexType>
         </xsd:element>

        <xsd:element name="GraphSelectionFactoryResponse" type="wsa:EndpointReferenceType" />
                <xsd:complexType name="GraphDoesNotExistFaultType"/>
                <xsd:element name="GraphDoesNotExistFault" type="wsdairdfs:GraphDoesNotExistFaultType"/>
                </xsd:schema>
        </wsdl:types>

<!-- WSDL MESSAGES ################################################ -->

    <!-- ############################################ -->
    <!-- ### GetCollectionPropertyDocument Messages ### -->
    <!-- ############################################ -->

        <wsdl:message name="GetCollectionPropertyDocumentRequest">
                <wsdl:part name="GetCollectionPropertyDocumentRequest"
                           element="wsdai:GetDataResourcePropertyDocumentRequest" />
        </wsdl:message>

        <wsdl:message name="GetCollectionPropertyDocumentResponse">
                <wsdl:part name="GetCollectionPropertyDocumentResponse"
                           element="wsdairdfs:RDFSCollectionPropertyDocument" />
        </wsdl:message>

        <!-- ########################### -->
        <!-- ### GetGraphs Messages ### -->
        <!-- ########################### -->

        <wsdl:message name="GetGraphsRequest">
                <wsdl:part name="GetGraphsRequest" element="wsdairdfs:GetGraphsRequest"/>
                </wsdl:message>

        <wsdl:message name="GetGraphsResponse">
                <wsdl:part name="GetGraphsResponse" element="wsdairdfs:GetGraphsResponse"/>
                </wsdl:message>

        <!-- ########################### -->
        <!-- ### AddGraphs Messages ### -->
        <!-- ########################### -->

        <wsdl:message name="AddGraphsRequest">
                <wsdl:part name="AddGraphsRequest" element="wsdairdfs:AddGraphsRequest"/>
        </wsdl:message>

        <wsdl:message name="AddGraphsResponse">
                <wsdl:part name="AddGraphsResponse" element="wsdairdfs:AddGraphsResponse"/>
```

```
	</wsdl:message>

	<!-- ############################ -->
	<!-- ### RemoveGraphs Messages ### -->
	<!-- ############################ -->

	<wsdl:message name="RemoveGraphsRequest">
		<wsdl:part name="RemoveGraphsRequest" element="wsdairdfs:RemoveGraphsRequest"/>
	</wsdl:message>

	<wsdl:message name="RemoveGraphsResponse">
		<wsdl:part name="RemoveGraphsResponse" element="wsdairdfs:RemoveGraphsResponse"/>
	</wsdl:message>

	<!-- ################################### -->
	<!-- ### GraphSelectionFactory Messages ### -->
	<!-- ################################### -->
	<wsdl:message name="GraphSelectionFactoryRequest">
		<wsdl:part name="GraphSelectionFactoryRequest"
			element="wsdairdfs:GraphSelectionFactoryRequest"/>
	</wsdl:message>

	<wsdl:message name="GraphSelectionFactoryResponse">
		<wsdl:part name="GraphSelectionFactoryResponse"
			element="wsdairdfs:GraphSelectionFactoryResponse"/>
	</wsdl:message>

	<wsdl:message name="GraphDoesNotExistFault">
		<wsdl:part name="GraphDoesNotExistFault"
			element="wsdairdfs:GraphDoesNotExistFault"/>
	</wsdl:message>


<!-- WSDL PORT TYPES ############################################ -->
	<wsdl:portType name="RDFSCollectionAccessPT">
		<wsdl:operation name="GetCollectionPropertyDocument">
			<wsdl:input name="GetCollectionPropertyDocumentRequest"
				message="wsdairdfs:GetCollectionPropertyDocumentRequest" />
			<wsdl:output name="GetCollectionPropertyDocumentResponse"
				message="wsdairdfs:GetCollectionPropertyDocumentResponse" />
			<wsdl:fault name="InvalidResourceNameFault"
				message="wsdai:InvalidResourceNameFault" />
			<wsdl:fault name="DataResourceUnavailableFault"
				message="wsdai:DataResourceUnavailableFault" />
			<wsdl:fault message="wsdai:NotAuthorizedFault"
				name="NotAuthorizedFault"/>
			<wsdl:fault message="wsdai:ServiceBusyFault"
```

```
                                        name="ServiceBusyFault" />
            </wsdl:operation>

            <wsdl:operation name="AddGraphs">
                    <wsdl:input message="wsdairdfs:AddGraphsRequest"/>
                    <wsdl:output message="wsdairdfs:AddGraphsResponse"/>
                    <wsdl:fault name="InvalidResourceNameFault"
                                        message="wsdai:InvalidResourceNameFault" />
                    <wsdl:fault name="DataResourceUnavailableFault"
                                        message="wsdai:DataResourceUnavailableFault" />
                    <wsdl:fault message="wsdai:NotAuthorizedFault"
                                        name="NotAuthorizedFault"/>
                    <wsdl:fault message="wsdai:ServiceBusyFault"
                                        name="ServiceBusyFault" />
                    </wsdl:operation>

            <wsdl:operation name="GetGraphs">
                    <wsdl:input message="wsdairdfs:GetGraphsRequest"/>
                    <wsdl:output message="wsdairdfs:GetGraphsResponse"/>
                    <wsdl:fault name="InvalidResourceNameFault"
                                        message="wsdai:InvalidResourceNameFault" />
                    <wsdl:fault name="DataResourceUnavailableFault"
                                        message="wsdai:DataResourceUnavailableFault" />
                    <wsdl:fault message="wsdai:NotAuthorizedFault"
                                        name="NotAuthorizedFault"/>
                    <wsdl:fault message="wsdai:ServiceBusyFault"
                                        name="ServiceBusyFault" />
        <wsdl:fault name="DatasetTooLargeFault"
                     message="wsdai:DatasetTooLargeFault"/>
            </wsdl:operation>

            <wsdl:operation name="RemoveGraphs">
                    <wsdl:input message="wsdairdfs:RemoveGraphsRequest"/>
                    <wsdl:output message="wsdairdfs:RemoveGraphsResponse"/>
                    <wsdl:fault name="InvalidResourceNameFault"
                                        message="wsdai:InvalidResourceNameFault" />
                    <wsdl:fault name="DataResourceUnavailableFault"
                                        message="wsdai:DataResourceUnavailableFault" />
                    <wsdl:fault message="wsdai:NotAuthorizedFault"
                                        name="NotAuthorizedFault"/>
                    <wsdl:fault message="wsdai:ServiceBusyFault"
                                        name="ServiceBusyFault" />
            </wsdl:operation>

    </wsdl:portType>

    <wsdl:portType name="RDFSCollectionFactoryPT">
```

```
            <wsdl:operation name="GraphSelectionFactory">
                  <wsdl:input message="wsdairdfs:GraphSelectionFactoryRequest"/>
                  <wsdl:output message="wsdairdfs:GraphSelectionFactoryResponse"/>
                  <wsdl:fault name="InvalidResourceNameFault"
                                    message="wsdai:InvalidResourceNameFault" />
                  <wsdl:fault name="DataResourceUnavailableFault"
                                    message="wsdai:DataResourceUnavailableFault" />
                  <wsdl:fault message="wsdai:NotAuthorizedFault"
                                    name="NotAuthorizedFault"/>
                  <wsdl:fault message="wsdai:ServiceBusyFault"
                                    name="ServiceBusyFault" />
                  <wsdl:fault name="GraphDoesNotExistFault"
                                    message="wsdairdfs:GraphDoesNotExistFault" />
                  <wsdl:fault message="wsdai:InvalidConfigurationDocumentFault"
                                    name="InvalidConfigurationDocumentFault" />
                  <wsdl:fault message="wsdai:InvalidPortTypeQNameFault"
                                    name="InvalidPortTypeQNameFault" />
            </wsdl:operation>
      </wsdl:portType>
</wsdl:definitions>
```

Appendix B.1 – SPARQL Property Document XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
    xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">
       <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI" schemaLocation="./wsdai_core_types.xsd"/>

       <!-- external graphs access flag -->
       <xsd:element name="ExternalGraphAccess" type="xsd:boolean"/>

       <!-- property documents -->
       <xsd:complexType name="SPARQLPropertyDocumentType">
              <xsd:complexContent>
                     <xsd:extension base="wsdai:PropertyDocumentType">
                            <xsd:sequence>
                                   <xsd:element ref="wsdairdfs:ExternalGraphAccess"/>
                            </xsd:sequence>
                     </xsd:extension>
              </xsd:complexContent>
       </xsd:complexType>
       <xsd:element name="SPARQLPropertyDocument" type="wsdairdfs:SPARQLPropertyDocumentType"/>
</xsd:schema>
```

Appendix B.2 – SPARQL WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdairdfs"
                          targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                          xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                          xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                     xmlns:wsa="http://www.w3.org/2005/08/addressing"
                          xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

<!-- WSDL IMPORTS ################################################ -->
       <wsdl:import location="./wsdai core porttypes.wsdl"
                          namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"/>
       <xsd:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="./ws-addressing-0805.xsd"/>


       <wsdl:types>
              <xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query" elementFormDefault="qualified">
                     <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI" schemaLocation="./wsdai core types.xsd" />
                     <xsd:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="./ws-addressing-0805.xsd" />
                     <xsd:include schemaLocation="./wsdairdfs sparqlpropertydocument types.xsd" />

              <!-- ############################ -->
              <!-- ### Common Message Types ### -->
              <!-- ############################ -->

                     <xsd:complexType name="SPARQLQueryRequestType">
                            <xsd:complexContent>
                                   <xsd:extension base="wsdai:ExpressionType">
                                          <xsd:sequence>
                                                 <xsd:element name="query" type="xsd:string" minOccurs="1" maxOccurs="1" />
                                                 <xsd:element name="default-graph-uri" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded" />
                                                 <xsd:element name="named-graph-uri" type="xsd:anyURI" minOccurs="0"
maxOccurs="unbounded" />
                                          </xsd:sequence>
                                   </xsd:extension>
                            </xsd:complexContent>
                     </xsd:complexType>

                     <xsd:element name="SPARQLQueryRequest" type="wsdairdfs:SPARQLQueryRequestType" abstract="true" />

                     <xsd:complexType name="SPARQLFaultType">
                            <xsd:element name="Detail" type="xsd:anyType" minOccurs="0"/>
                     </xsd:complexType>
```

```
                        <xsd:element name="SPARQLFault" type="wsdairdfs:SPARQLFaultType"/>

                        <xsd:complexType name="ExternalGraphFaultType"/>
                        <xsd:element name="ExternalGraphFault" type="wsdairdfs:ExternalGraphFaultType"/>

                        <!-- ################################## -->
                        <!-- ### SPARQLExecute Message Types ### -->
                        <!-- ################################## -->

                        <xsd:element name="SPARQLExecuteRequest">
                               <xsd:complexType>
                                      <xsd:complexContent>
                                             <xsd:extension base="wsdai:RequestType">
                                                    <xsd:sequence>
                                                           <xsd:element ref="wsdairdfs:SPARQLQueryRequest"  minOccurs="1"
maxOccurs="1"/>
                                                    </xsd:sequence>
                                             </xsd:extension>
                                      </xsd:complexContent>
                               </xsd:complexType>
                        </xsd:element>

                        <xsd:element name="SPARQLExecuteResponse">
                               <xsd:complexType>
                                      <xsd:sequence>
                                             <xsd:element ref="wsdai:Dataset"  minOccurs="0" maxOccurs="1"/>
                                      </xsd:sequence>
                               </xsd:complexType>
                        </xsd:element>

                <!-- ######################################### -->
                <!-- ### SPARQLExecuteFactory Message Types ### -->
                <!-- ######################################### -->

                        <xsd:element name="SPARQLExecuteFactoryRequest">
                               <xsd:complexType>
                                      <xsd:complexContent>
                                             <xsd:extension base="wsdai:FactoryRequestType">
                                                    <xsd:sequence>
                                                           <xsd:element ref="wsdairdfs:SPARQLQueryRequest"  minOccurs="1"
maxOccurs="1"/>
                                                    </xsd:sequence>
                                             </xsd:extension>
                                      </xsd:complexContent>
                               </xsd:complexType>
                        </xsd:element>
```

```
                    <xsd:element name="SPARQLExecuteFactoryResponse" type="wsa:EndpointReferenceType" />

            </xsd:schema>
        </wsdl:types>

<!-- WSDL MESSAGES ################################################ -->
        <wsdl:message name="SPARQLFault">
            <wsdl:part name="SPARQLFault" element="wsdairdfs:SPARQLFault"/>
        </wsdl:message>

        <wsdl:message name="ExternalGraphFault">
            <wsdl:part name="ExternalGraphFault" element="wsdairdfs:ExternalGraphFault"/>
        </wsdl:message>

        <!-- ############################################## -->
        <!-- ###   GetSPARQLPropertyDocument Messages   ### -->
        <!-- ############################################## -->
        <wsdl:message name="GetSPARQLPropertyDocumentRequest">
            <wsdl:part name="GetSPARQLPropertyDocumentRequest" element="wsdai:GetDataResourcePropertyDocumentRequest" />
        </wsdl:message>

        <wsdl:message name="GetSPARQLPropertyDocumentResponse">
            <wsdl:part name="GetSPARQLPropertyDocumentResponse" element="wsdairdfs:SPARQLPropertyDocument" />
        </wsdl:message>

        <!-- ############################## -->
        <!-- ### SPARQLExecute Messages ### -->
        <!-- ############################## -->
        <wsdl:message name="SPARQLExecuteRequest">
            <wsdl:part name="SPARQLExecuteRequest" element="wsdairdfs:SPARQLExecuteRequest"/>
        </wsdl:message>

        <wsdl:message name="SPARQLExecuteResponse">
            <wsdl:part name="SPARQLExecuteResponse" element="wsdairdfs:SPARQLExecuteResponse"/>
        </wsdl:message>

        <!-- ################################### -->
        <!-- ### SPARQLExecuteFactory Messages ### -->
        <!-- ################################### -->
        <wsdl:message name="SPARQLExecuteFactoryRequest">
            <wsdl:part name="SPARQLExecuteFactoryRequest" element="wsdairdfs:SPARQLExecuteFactoryRequest"/>
        </wsdl:message>

        <wsdl:message name="SPARQLExecuteFactoryResponse">
            <wsdl:part name="SPARQLExecuteFactoryResponse" element="wsdairdfs:SPARQLExecuteFactoryResponse"/>
        </wsdl:message>
```

```
<!-- WSDL PORT TYPES ############################################ -->
       <wsdl:portType name="SPARQLAccessPT">
              <wsdl:operation name="GetSPARQLPropertyDocument">
                     <wsdl:input name="GetSPARQLPropertyDocumentRequest" message="wsdairdfs:GetSPARQLPropertyDocumentRequest" />
                     <wsdl:output name="GetSPARQLPropertyDocumentResponse" message="wsdairdfs:GetSPARQLPropertyDocumentResponse" />
                     <wsdl:fault name="InvalidResourceNameFault" message="wsdai:InvalidResourceNameFault" />
                     <wsdl:fault name="DataResourceUnavailableFault" message="wsdai:DataResourceUnavailableFault" />
                     <wsdl:fault message="wsdai:NotAuthorizedFault" name="NotAuthorizedFault"/>
                     <wsdl:fault message="wsdai:ServiceBusyFault" name="ServiceBusyFault" />
              </wsdl:operation>

              <wsdl:operation name="SPARQLExecute">
                     <wsdl:input message="wsdairdfs:SPARQLExecuteRequest"/>
                     <wsdl:output message="wsdairdfs:SPARQLExecuteResponse"/>
                     <wsdl:fault name="InvalidResourceNameFault" message="wsdai:InvalidResourceNameFault" />
                     <wsdl:fault name="DataResourceUnavailableFault" message="wsdai:DataResourceUnavailableFault" />
                      <wsdl:fault message="wsdai:InvalidLanguageFault" name="InvalidLanguageFault" />                    <wsdl:fault
message="wsdai:InvalidExpressionFault" name="InvalidExpressionFault"/>
                     <wsdl:fault message="wsdai:InvalidDatasetFormatFault" name="InvalidDatasetFormatFault"/>
                     <wsdl:fault message="wsdai:NotAuthorizedFault" name="NotAuthorizedFault"/>
                     <wsdl:fault message="wsdai:ServiceBusyFault" name="ServiceBusyFault" />
                     <wsdl:fault name="SPARQLFault" message="wsdairdfs:SPARQLFault" />
                     <wsdl:fault name="ExternalGraphFault" message="wsdairdfs:ExternalGraphFault" />
            <wsdl:fault name="DatasetTooLargeFault" message="wsdai:DatasetTooLargeFault"/>
              </wsdl:operation>
       </wsdl:portType>

       <wsdl:portType name="SPARQLFactoryPT">
              <wsdl:operation name="SPARQLExecuteFactory">
                     <wsdl:input message="wsdairdfs:SPARQLExecuteFactoryRequest"/>
                     <wsdl:output message="wsdairdfs:SPARQLExecuteFactoryResponse"/>
                     <wsdl:fault name="InvalidResourceNameFault" message="wsdai:InvalidResourceNameFault" />
                     <wsdl:fault name="DataResourceUnavailableFault" message="wsdai:DataResourceUnavailableFault" />
                     <wsdl:fault message="wsdai:InvalidLanguageFault" name="InvalidLanguageFault" />
                     <wsdl:fault message="wsdai:InvalidExpressionFault" name="InvalidExpressionFault"/>
                     <wsdl:fault message="wsdai:InvalidPortTypeQNameFault" name="InvalidPortTypeQNameFault"/>
                     <wsdl:fault message="wsdai:InvalidConfigurationDocumentFault" name="InvalidConfigurationDocumentFault"/>
                     <wsdl:fault message="wsdai:NotAuthorizedFault" name="NotAuthorizedFault"/>
                     <wsdl:fault message="wsdai:ServiceBusyFault" name="ServiceBusyFault" />
                     <wsdl:fault name="SPARQLFault" message="wsdairdfs:SPARQLFault" />
              </wsdl:operation>
       </wsdl:portType>
</wsdl:definitions>
```

Appendix C.1 –SPARQLItemsSet XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
```

```
        elementFormDefault="qualified"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
               xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

       <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                           schemaLocation="./wsdai_core_types.xsd" />

       <xsd:element name="NumberOfItems" type="xsd:unsignedLong"/>

<!-- property documents -->

       <xsd:complexType name="SPARQLItemsSetPropertyDocumentType">
              <xsd:complexContent>
                     <xsd:extension base="wsdai:PropertyDocumentType">
                           <xsd:sequence>
                                  <xsd:element ref="wsdairdfs:NumberOfItems" />
                           </xsd:sequence>
                     </xsd:extension>
              </xsd:complexContent>
       </xsd:complexType>
    <xsd:element name="SPARQLItemsSetPropertyDocument" type="wsdairdfs:SPARQLItemsSetPropertyDocumentType"/>
</xsd:schema>
```

Appendix C.2 – SPARQLResultSet WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdairdfs"
                           targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                           xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                           xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                           xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                           xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

<!-- WSDL IMPORTS ################################################ -->
       <wsdl:import location="./wsdai_core_porttypes.wsdl"
                           namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"/>

<!-- WSDL TYPES ################################################## -->
       <wsdl:types>
              <xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                                  elementFormDefault="qualified">
                     <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                                         schemaLocation="./wsdai_core_types.xsd" />
                     <xsd:include schemaLocation="./wsdairdfs_sparqlitemsset_types.xsd" />
```

```
                        <!-- ##################################### -->
                        <!-- ### SPARQLResultsSet Message Types ### -->
                        <!-- ##################################### -->

                        <xsd:complexType name="InvalidStartPositionFaultType"/>
                        <xsd:element name="InvalidStartPositionFault" type="wsdairdfs:InvalidStartPositionFaultType"/>
                        <xsd:complexType name="InvalidCountFaultType"/>
                        <xsd:element name="InvalidCountFault" type="wsdairdfs:InvalidCountFaultType"/>

                        <xsd:element name="GetResultsRequest">
                                <xsd:complexType>
                                        <xsd:complexContent>
                                                <xsd:extension base="wsdai:RequestType">
                                                        <xsd:sequence>
                                                                <xsd:element name="StartPosition" type="xsd:integer"/>
                                                                <xsd:element name="ResultCount" type="xsd:integer"/>
                                                        </xsd:sequence>
                                                </xsd:extension>
                                        </xsd:complexContent>
                                </xsd:complexType>
                        </xsd:element>

                        <xsd:element name="GetResultsResponse">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element ref="wsdai:Dataset"  minOccurs="1" maxOccurs="1"/>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                </xsd:schema>
        </wsdl:types>

<!-- WSDL MESSAGES ################################################ -->

        <!-- ################################################ -->
        <!-- ### GetSPARQLItemsSetPropertyDocument Messages ### -->
        <!-- ################################################ -->

        <wsdl:message name="GetSPARQLItemsSetPropertyDocumentRequest">
                <wsdl:part name="GetSPARQLItemsSetPropertyDocumentRequest"
                        element="wsdai:GetDataResourcePropertyDocumentRequest" />
        </wsdl:message>

        <wsdl:message name="GetSPARQLItemsSetPropertyDocumentResponse">
                <wsdl:part name="GetSPARQLItemsSetPropertyDocumentResponse"
                        element="wsdairdfs:SPARQLItemsSetPropertyDocument" />
```

```
        </wsdl:message>

        <!-- ########################## -->
        <!-- ### GetResults Messages ### -->
        <!-- ########################## -->

        <wsdl:message name="GetResultsRequest">
                <wsdl:part name="GetResultsRequest" element="wsdairdfs:GetResultsRequest"/>
        </wsdl:message>

        <wsdl:message name="GetResultsResponse">
                <wsdl:part name="GetResultsResponse" element="wsdairdfs:GetResultsResponse"/>
        </wsdl:message>

        <wsdl:message name="InvalidStartPositionFault">
                <wsdl:part name="InvalidStartPositionFault"
                                element="wsdairdfs:InvalidStartPositionFault"/>
        </wsdl:message>

        <wsdl:message name="InvalidCountFault">
                <wsdl:part name="InvalidCountFault"
                                element="wsdairdfs:InvalidCountFault"/>
        </wsdl:message>

<!-- WSDL PORT TYPES ############################################ -->

        <wsdl:portType name="SPARQLQueryResultsAccessPT">
                <wsdl:operation name="GetSPARQLItemsSetPropertyDocument">
                        <wsdl:input name="GetSPARQLItemsSetPropertyDocumentRequest"
                                        message="wsdairdfs:GetSPARQLItemsSetPropertyDocumentRequest" />
                        <wsdl:output name="GetSPARQLItemsSetPropertyDocumentResponse"
                                        message="wsdairdfs:GetSPARQLItemsSetPropertyDocumentResponse" />
                        <wsdl:fault name="InvalidResourceNameFault"
                                        message="wsdai:InvalidResourceNameFault" />
                        <wsdl:fault message="wsdai:NotAuthorizedFault"
                                        name="NotAuthorizedFault"/>
                        <wsdl:fault name="DataResourceUnavailableFault"
                                        message="wsdai:DataResourceUnavailableFault" />
                        <wsdl:fault name="ServiceBusyFault"
                                        message="wsdai:ServiceBusyFault" />
                </wsdl:operation>

                <wsdl:operation name="GetResults">
                        <wsdl:input message="wsdairdfs:GetResultsRequest"/>
                        <wsdl:output message="wsdairdfs:GetResultsResponse"/>
                        <wsdl:fault name="InvalidResourceNameFault"
                                        message="wsdai:InvalidResourceNameFault" />
```

```
                        <wsdl:fault name="DataResourceUnavailableFault"
                                    message="wsdai:DataResourceUnavailableFault" />
                        <wsdl:fault message="wsdai:InvalidDatasetFormatFault"
                                    name="InvalidDatasetFormatFault"/>
                        <wsdl:fault message="wsdai:NotAuthorizedFault"
                                    name="NotAuthorizedFault"/>
                        <wsdl:fault message="wsdai:ServiceBusyFault"
                                    name="ServiceBusyFault" />
                        <wsdl:fault name="InvalidStartPositionFault"
                                    message="wsdairdfs:InvalidStartPositionFault" />
                        <wsdl:fault name="InvalidCountFault"
                                    message="wsdairdfs:InvalidCountFault" />
              <wsdl:fault name="DatasetTooLargeFault"
                          message="wsdai:DatasetTooLargeFault"/>
                 </wsdl:operation>

        </wsdl:portType>

</wsdl:definitions>
```

Appendix C.3 – SPARQLTriplesSet WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdairdfs"
                             targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                             xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                             xmlns:wsdai="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                             xmlns:wsdairdfs="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query">

<!-- WSDL IMPORTS ################################################ -->
        <wsdl:import location="./wsdai_core_porttypes.wsdl"
                             namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"/>

<!-- WSDL TYPES ################################################## -->
        <wsdl:types>
                <xsd:schema targetNamespace="http://www.ogf.org/namespaces/2006/12/WS-DAI-RDFS/Query"
                                    elementFormDefault="qualified">
                        <xsd:import namespace="http://www.ggf.org/namespaces/2005/12/WS-DAI"
                                    schemaLocation="./wsdai_core_types.xsd" />
                        <xsd:include schemaLocation="./wsdairdfs_sparqlitemsset_types.xsd" />

                        <!-- ################################### -->
                        <!-- ### SPARQLTriplesSet Message Types ### -->
                        <!-- ################################### -->

                        <xsd:complexType name="InvalidStartPositionFaultType"/>
                        <xsd:element name="InvalidStartPositionFault" type="wsdairdfs:InvalidStartPositionFaultType"/>
```

```
                        <xsd:complexType name="InvalidCountFaultType"/>
                        <xsd:element name="InvalidCountFault" type="wsdairdfs:InvalidCountFaultType"/>

                        <xsd:element name="GetTriplesRequest">
                                <xsd:complexType>
                                        <xsd:complexContent>
                                                <xsd:extension base="wsdai:RequestType">
                                                        <xsd:sequence>
                                                                <xsd:element name="StartPosition" type="xsd:integer"/>
                                                                <xsd:element name="Count" type="xsd:integer"/>
                                                        </xsd:sequence>
                                                </xsd:extension>
                                        </xsd:complexContent>
                                </xsd:complexType>
                        </xsd:element>

                        <xsd:element name="GetTriplesResponse">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element ref="wsdai:Dataset"  minOccurs="1" maxOccurs="1"/>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                </xsd:schema>
        </wsdl:types>

<!-- WSDL MESSAGES ################################################ -->

        <!-- ################################################ -->
        <!-- ### GetSPARQLItemsSetPropertyDocument Messages ### -->
        <!-- ################################################ -->

        <wsdl:message name="GetSPARQLItemsSetPropertyDocumentRequest">
                <wsdl:part name="GetSPARQLItemsSetPropertyDocumentRequest"
                        element="wsdai:GetDataResourcePropertyDocumentRequest" />
        </wsdl:message>

        <wsdl:message name="GetSPARQLItemsSetPropertyDocumentResponse">
                <wsdl:part name="GetSPARQLItemsSetPropertyDocumentResponse"
                        element="wsdairdfs:SPARQLItemsSetPropertyDocument" />
        </wsdl:message>

        <!-- ########################### -->
        <!-- ### GetTriples Messages ### -->
        <!-- ########################### -->
        <wsdl:message name="GetTriplesRequest">
                <wsdl:part name="GetTriplesRequest" element="wsdairdfs:GetTriplesRequest"/>
```

```
        </wsdl:message>

        <wsdl:message name="GetTriplesResponse">
                <wsdl:part name="GetTriplesResponse" element="wsdairdfs:GetTriplesResponse"/>
        </wsdl:message>

        <wsdl:message name="InvalidStartPositionFault">
                <wsdl:part name="InvalidStartPositionFault"
                                element="wsdairdfs:InvalidStartPositionFault"/>
        </wsdl:message>

        <wsdl:message name="InvalidCountFault">
                <wsdl:part name="InvalidCountFault"
                                element="wsdairdfs:InvalidCountFault"/>
        </wsdl:message>

<!-- WSDL PORT TYPES ############################################ -->
        <wsdl:portType name="SPARQLTriplesSetAccessPT">
                <wsdl:operation name="GetSPARQLItemsSetPropertyDocument">
                        <wsdl:input name="GetSPARQLItemsSetPropertyDocumentRequest"
                                                message="wsdairdfs:GetSPARQLItemsSetPropertyDocumentRequest" />
                        <wsdl:output name="GetSPARQLItemsSetPropertyDocumentResponse"
                                                message="wsdairdfs:GetSPARQLItemsSetPropertyDocumentResponse" />
                        <wsdl:fault name="InvalidResourceNameFault"
                                                message="wsdai:InvalidResourceNameFault" />
                        <wsdl:fault message="wsdai:NotAuthorizedFault"
                                                name="NotAuthorizedFault"/>
                        <wsdl:fault name="DataResourceUnavailableFault"
                                                message="wsdai:DataResourceUnavailableFault" />
                        <wsdl:fault name="ServiceBusyFault"
                                                message="wsdai:ServiceBusyFault" />
                </wsdl:operation>

                <wsdl:operation name="GetTriples">
                        <wsdl:input message="wsdairdfs:GetTriplesRequest"/>
                        <wsdl:output message="wsdairdfs:GetTriplesResponse"/>
                        <wsdl:fault name="InvalidResourceNameFault"
                                                message="wsdai:InvalidResourceNameFault" />
                        <wsdl:fault name="DataResourceUnavailableFault"
                                                message="wsdai:DataResourceUnavailableFault" />
                        <wsdl:fault message="wsdai:InvalidDatasetFormatFault"
                                                name="InvalidDatasetFormatFault"/>
                        <wsdl:fault message="wsdai:NotAuthorizedFault"
                                                name="NotAuthorizedFault"/>
                        <wsdl:fault message="wsdai:ServiceBusyFault"
                                                name="ServiceBusyFault" />
                        <wsdl:fault name="InvalidStartPositionFault"
```

```
                                        message="wsdairdfs:InvalidStartPositionFault" />
                    <wsdl:fault name="InvalidCountFault"
                                        message="wsdairdfs:InvalidCountFault" />
            <wsdl:fault name="DatasetTooLargeFault"
                        message="wsdai:DatasetTooLargeFault"/>
               </wsdl:operation>

        </wsdl:portType>
```

</wsdl:definitions>