

GWD-R
GGF DAIS Working Group

Editors:
Mario Antonioletti, University of Edinburgh
Malcolm Atkinson, NESC
Amy Krause, University of Edinburgh
Simon Laws, IBM
Susan Malaika, IBM
Norman W Paton, University of Manchester
Dave Pearson, Oracle
Greg Riccardi, Florida State University

Category: INFORMATIONAL

May 21, 2004

Web Services Data Access and Integration (WS-DAI)

Status of This Memo

This memo provides information regarding the specification of service based interfaces to data resources. The specification is presently a draft for discussion. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

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

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 GGF (Global Grid Forum) Open Grid Services Architecture (OGSA), and its associated specifications, defines consistent interfaces through web services between components of the grid infrastructure. Both the web and grid communities would benefit from the provision of consistent and agreed web service interfaces for data resources and the systems that manage them.

This document *Web Services Data Access and Integration (WS-DAI)*, presents a specification for a collection of generic data interfaces that can be extended to support specific kinds of data resources, such as relational databases, XML repositories, object databases, or files. The document builds on the GGF OGSA Data Services [Data Services] document in classifying functionality as pertaining to data description, data access, data factory, or data management.

This document is presented for discussion within the GGF Database Access and Integration Services (DAIS) Working Group, with a view to the document evolving to become a proposed recommendation. There are several respects in which the current proposal is incomplete, but it is hoped that the material included is sufficient to allow an informed discussion to take place concerning both its form and substance.

Related DAIS specifications define how specific data resources and systems can be described and manipulated through web services. The DAIS specifications form part of a broader activity within the GGF to develop OGSA. The DAIS specifications can be applied in regular web services environments or as part of a grid fabric.

Contents

Abstract.....	1
1. Introduction.....	4
1.1 Specification Scope.....	4
1.2 Specification Organization.....	4
2. Notational Conventions.....	5
3. Terminology.....	5
3.1 Data Resource.....	5
3.2 Data Service.....	6
3.3 Consumer.....	6
3.4 Data Set.....	6
4. Concepts.....	6
4.1 Data Service Model.....	6
4.2 Interface.....	7
4.3 Interface Composition.....	7
4.4 Identity.....	7
4.5 Informational Properties.....	7
4.6 Behavioral Properties.....	8
4.7 Direct Data Access.....	8
4.8 Derived Data Access.....	8
4.9 Subscription Based Data Access.....	8
4.10 Lifetime.....	8
4.11 Sessions.....	9
4.12 Access Control.....	9
4.13 Operation Validity.....	9
5. Data Description.....	9
5.1 Name.....	9
5.2 Description.....	9
6. Data Access.....	10
6.1 Message Patterns.....	10
6.2 Behavioral Properties.....	11
6.3 Messages.....	13
7. Data Factory.....	13
7.1 Message Patterns.....	14
7.2 Behavioral Properties.....	15
7.3 Messages.....	15
8. Data Management.....	15
8.1 Behavioral Properties.....	15
8.2 Messages.....	15
9. Mapping to WSDL.....	16
9.1 Data Services.....	16
9.2 Data Resources.....	16
9.3 Data Description.....	17
9.4 Data Access.....	17
9.5 Data Factory.....	17
9.6 Data Management.....	17
9.7 Faults.....	17
9.8 Sessions.....	18
9.9 Lifetime.....	18
10. Security Considerations.....	18
11. Conclusions.....	18
Editor Information.....	18
Contributors.....	19
Acknowledgements.....	19
Intellectual Property Statement.....	19

Full Copyright Notice20

References.....20

Appendix A – XML Schema23

Appendix B – WSDL.....26

1. Introduction

Data access plays a central role for many types of Grid applications. By data access we mean the retrieval, manipulation and insertion of data, which may be stored using a range of different formats and infrastructures. Grid data access requires a flexible framework for handling data requests to a data resource that is to be integrated within a Grid fabric as defined by the Open Grid Services Architecture (OGSA) [OGSA] of the Global Grid Forum (GGF).

This document provides a specification for a collection of generic Grid data access interfaces that are made available as web services. The base framework for these interfaces is defined in the *OGSA Data Services* [Data Services] document. The interfaces described here are categorized according to the support they provide for:

- Data description: provides metadata about pertinent characteristics of a data resource that a service may wish to expose as well as any associated properties that affect the interaction between a service and the data resource.
- Data access: provide access to data through a service interface.
- Data factory: exposing derived data through a service interface.
- Data management: manages the relationship between a service and the data resource that it exposes.

This document should be read in conjunction with the *OGSA Data Services* proposal.

This specification aims to provide a framework for Data Service interfaces and properties. It stops short of describing specialised interfaces. The framework described here may then be extended to define interfaces to access particular types of data, as is done in the current proposals to access relational [WS-DAIR] and XML [WS-DAIX] representations of data. Future specifications for accessing other specialised forms of data, for example, files or object databases, may also extend the base set of interfaces defined in this document.

1.1 Specification Scope

The *OGSA Data Services* document [Data Services] introduces the `DataAccess`, `DataFactory` and `DataManagement` interface groupings as well as `DataDescription` properties. These classifications are adopted and extended by DAIS.

In this document the DAIS Working Group specifies a Data Service in terms of the base `DataAccess` and `DataFactory` interfaces and base `DataDescription` properties which it may implement.

It is not the intention of the DAIS working group to define new query languages or data models. `DataAccess` interfaces are therefore described in terms of existing language interfaces supported by the underlying Data Resource to which the service is providing access.

The term data management generally refers to any number of activities related to describing, storing and providing access to data. In this document `DataManagement` refers to the management of the relationship between the web service interface and the Data Resource that stores the data. `DataManagement` has not yet been fully considered in this specification. Section 8 is therefore a work in progress.

1.2 Specification Organization

This specification separates the abstract model of a Data Service from its operational representation. The abstract model is described using the terminology defined in Section 3 and employs the concepts introduced in Section 4. Sections 5, 6, 7 and 8 present the Data

Description, Data Access, Data Factory and Data Management aspects of the abstract model respectively.

A mapping of the abstract model to the web services resource framework (WSRF) is described in Section 9.

Section 10 discusses security. Section 11 draws conclusions from this specification exercise.

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/>`).

Italicised element names are used when the element is intended to be specified by subsequent DAIS specifications.

When patterns of messages are described the layout of the XML of each message is presented, as opposed to the XML schema. The following notation is used to indicate cardinality of XML elements in these cases:

- * 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 *OGSA Data Services* document [Data Services]. In particular the terms Data Service, Data Resource and Data Set are used. The OGSA Data Services document is still evolving and this terminology is likely to change in future versions of the DAIS Working Group specifications.

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
http	http://www.w3.org/2002/06/wsdl/http
wsdl	http://schemas.xmlsoap.org/wsdl/
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
wsdai	http://www.ggf.org/namespaces/2004/05/WS-DAI

3. Terminology

3.1 Data Resource

A Data Resource is any system that can act as a source or sink of data. Examples of Data Resources include relational or XML databases, file systems, sensor networks, etc. We assert that data in the Grid will still generally be managed using existing technologies such as relational

databases or file systems. Moving data to new storage technology to allow a Grid application access in most cases will not be an option.

An existing Data Resource will already provide consumers with mechanisms for accessing stored data. The DAIS specifications provide a common service oriented treatment for Data Resources that exposes these mechanisms in a manner that supports integration into an OGSA based Grid fabric.

3.2 Data Service

A Data Service is simply a web service that implements one or more of the DAIS specified interfaces to provide access to Data Resources. It is not the intention of DAIS to define new universal query languages or data models. The specifications of the DAIS working group provide a web service enabled flexible data access framework exposing existing data access techniques already available in a Data Resource and using other relevant specifications as required by the Grid.

3.3 Consumer

A Consumer is an application that exploits the interface provided by a Data Service in order to access a Data Resource. A Data Service may act as a Consumer to another Data Service.

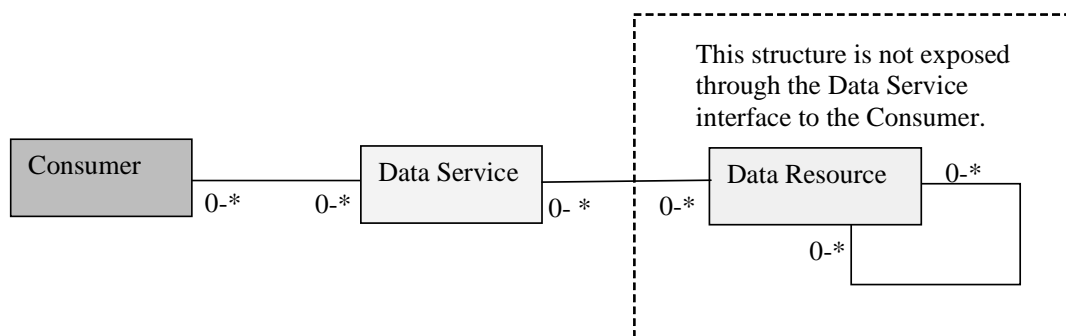
3.4 Data Set

A Data Set is an encoding of data suitable for externalization outside a Data Service, for example, as an XML document or as a binary stream. The concept of a Data Set is introduced to describe data as it appears in the messages passing to and from Data Services, i.e. between the Consumer and the Data Service.

4. Concepts

4.1 Data Service Model

The focus of this specification is on defining base Data Service interfaces. These interfaces follow the model described in the *OGSA Data Services* document [Data Services]. In this model there can be a many to many relationship between Consumers and Data Services, and between Data Services and Data Resources.



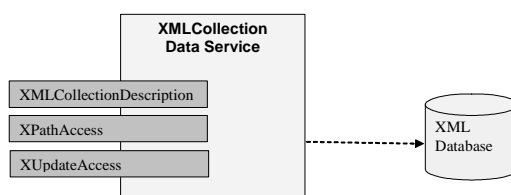
A Data Service presents a Consumer with an interface to a Data Resource. A Data Resource can have arbitrary complexity, for example, a file on an NFS mounted file system or a federation of relational databases. A Consumer is not typically exposed to this complexity and operates within the bounds and semantics of the interface provided by the Data Service.

4.2 Interface

The word interface refers to the collections of messages and XML structures that describe the ways in which a consumer can validly interact with a Data Service. It is not intended to refer specifically to the proposed use of the word interface found in the current working draft of the WSDL 2.0 specification although this may be an appropriate mapping in the future.

4.3 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. For example:



Here a Data Service provides XPathAccess and XUpdateAccess interfaces for an XMLCollection Data Service that, in this case, is associated with an XML Database.

4.4 Identity

A Data Service is a web service and is located in the global context in the usual way using a URL. The Data Resource **MUST** be identifiable within the context of the Data Service and **MAY** be identifiable in a wider context.

This specification does not define a Data Resource naming scheme and gives no guidance on the information that might be used to locate a suitable Data Service or Data Resource, for example, via a registry.

A descriptive element through which the name of a Data Resource **MAY** be identified is provided as part of the base Data Description but this specification does not describe the type of the name or the scope in which it is considered valid. If a Name property is provided it **MUST** be read only and **MUST NOT** be settable by the Consumer exception at the time the relationship between Data Service and Data Resource is created.

Name information, along with any associated metadata, **MAY** be stored in a registry to aid Data Service and Data Resource discovery.

The mapping section in each of the specifications describes the mechanism used to identify a Data Resource within a Data Service.

4.5 Informational Properties

Properties describe the characteristics of a Data Resource as well as the Data Service's relationship with that Data Resource. These properties are XML elements. They **SHOULD** be made available through the Data Service associated with a Data Resource. They **SHOULD** be set when a Data Resource is associated with a Data Service. Their values **MAY** change over time. They **MAY NOT** be settable by the consumer.

The structure of individual properties depends on the Data Service in question.

The Data Description section of each DAIS specification collects together property definitions. The mapping section provides mechanisms for accessing this information.

4.6 Behavioral Properties

Properties also describe the behavioral characteristics of a Data Service. Properties SHOULD be accessible as XML elements. They SHOULD be made available through a Data Service associated with a given Data Resource. Behavioral properties SHOULD be set when a Data Resource is associated with a Data Service. Behavioral property values SHOULD NOT change over time. The modification of behavioral properties by the Consumer MAY be supported after Data Service creation.

The structure and valid states of any particular property are dependent on the Data Service and the particular behavior that the property represents.

In the DAIS specifications, behavioral properties are described alongside the Data Access and Data Factory operations, the behavior of which they describe.

There is work underway in the GGF GRAAP working group to define the WS Agreement specification [WS-Agreement]. This formally specifies a mechanism for advertising and negotiating agreements which can describe service behavior. The DAIS working group is tracking the development of this specification with a view to representing behavioral terms as agreement documents.

4.7 Direct Data Access

In this specification direct data access means that a consumer can expect a direct response to the requests made to a Data Service. For example, passing an XPath query message to a Data Service will result in a response message containing a set of XML fragments – this is considered to be direct data access.

4.8 Derived Data Access

In this specification derived data access means that a Consumer does not expect the results as the response to a request made to a Data Service. The request to access data will be processed by the Data Service and Data Resource with the results being made available to the Consumer at a later time, usually through a different Data Service and interface.

For example, in this mode passing an SQL query message to a Data Service will result in a reference to another Data Service (and possibly another Data Resource) being produced that allows access to the result of the original query. This allows results to be held at the service side for further processing thus minimizing unnecessary data movement. The consumer may then use this new reference to retrieve the results or allow for further processing to take place at the server side when the results become available.

The structure of Data Service references is specific to each mapping.

4.9 Subscription Based Data Access

The DAIS specification does not consider subscription based data access, where the consumer supplies a profile describing the data of interest and the conditions under which it will be delivered. Work is ongoing in the GGF Information Dissemination working group to specify this model.

4.10 Lifetime

The notion proposed by the OGSA Data Services document of accessing Data Resources through Data Services implies the need to control the lifetime of the relationship between Data Service and Data Resource. The DAIS specifications propose that the relationship is formed by

operations that implement the factory pattern. The DAIS specifications do not describe how the relationship ends.

4.11 Sessions

The DAIS specifications do not describe how multiple requests to a Data Service are correlated either for single or multiple consumers, or for single or multiple requests. This is left to other proposed web service specifications. For example, WS Coordination [WS-Coordination] or WS Context [WS-Context]

4.12 Access Control

The possibility of many client processes accessing a Data Service interface, possibly concurrently, is assumed to be the default situation.

Access to a Data Service and the data it represents may be denied, where appropriate, using suitable access mechanisms either at the service or the underlying Data Resource. For example, a system representing a relational database may wish to restrict access to the Data Services representing RowSets to the client that submitted the query that created the RowSets.

The requirement to pass security related information is common with many other specifications. The DAIS working group expects this requirement to be satisfied using other specifications such as WS Security [WS-Security].

4.13 Operation Validity

The DAIS working group specifications describe messages and properties in accordance with the type of interface being presented. The appearance of an operation in a portType does not guarantee that it may be called validly in any particular situation. Faults are provided to notify the caller that an operation could not be completed successfully.

5. Data Description

Data Description contains XML structures that describe the informational properties of a Data Resource. The informational properties defined here **MUST** appear in all Data Services that implement Data Description. DAIS realizations based on this specification will extend the list of informational properties as required.

The mapping section describes how these elements are made available in WSDL.

5.1 Name

```
<xsd:element name="Name" >
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:any minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

/wsdai:Name

The name associated with the Data Resource represented by the Data Service. The scope and structure of this name is not defined.

5.2 Description

```
<xsd:element name="Description" >
  <xsd:complexType mixed="true">
```

```

<xsd:sequence>
  <xsd:any minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

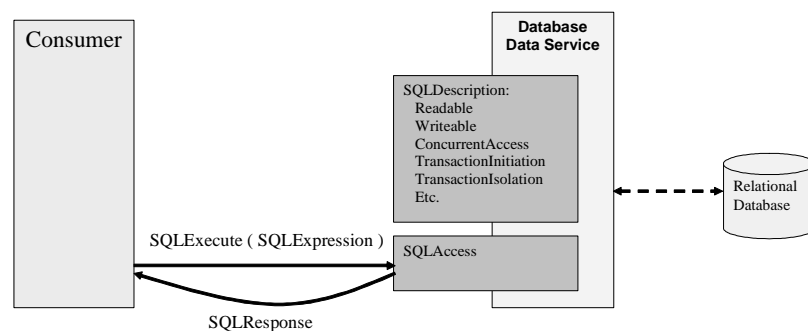
```

/wsdai:Description

A free format textual description of the Data Resource as represented by a Data Service.

6. Data Access

Data Access collects together messages that directly access and modify the data represented by a Data Service along with the behavioral properties which describe the behavior of these access messages. For example,



The Database Data Service, in the diagram above, implements the SQLAccess messages and exposes the SQLDescription informational properties; more details about these properties can be found in [WS-DAIR]. In this example, a Consumer uses the SQLExecute message to submit an SQL expression. The associated response message will contain the results of the SQL execute request. When the SQL expression used is a SELECT statement, the SQL response will contain a RowSet.

The behavior of the Database Data Service is controlled, in part, by its behavioral properties. These properties are associated with Data Access and are made accessible as part of the relational description properties. The behavioral properties will include the properties defined in this specification. For example, ConcurrentAccess will be set to TRUE if the Database Data Service is able to process messages from more than one consumer at a time.

All of the behavioral properties defined in this specification SHOULD appear in all Data Services that implement Data Access. DAIS realizations based on this specification MAY extend the list of terms and messages as required and MUST define one or more access messages.

6.1 Message Patterns

DAIS specified Data Access interfaces support messages that allow Data Sets to be passed into or retrieved from a Data Service and describe the messages that provide direct data access from a Data Service.

The structure of a direct data access request message xml instance is:

```

<wsdai:RequestMessage>
  <wsdai:RequestDocument />
  <wsdai:ResponseFormat />?
</wsdai:RequestMessage>

```

/wsdai:RequestMessage

This is the root element for a request message. The type of this element is specific to each message.

/wsdai:RequestMessage/RequestDocument

This element contains the request expression. The structure of this document is specific to the expression being used. The name of this element implies the language used by the expression.

/wsdai:RequestMessage/ResponseFormat

An optional element that can be used to define the format of the response message. This element **MUST** contain a QName from the set that appears in the *RequestMessageResponseTypeList* informational property element. When only one QName is advertise this element **MAY** be omitted in which case the format of the response message will follow that of the type reference by the advertised QName.

The structure of a direct access response message is:

```
<wsdai:ResponseMessage>
  Data goes here formatted according to the response format parameter of
  request message
</wsdai:ResponseMessage>
```

The structure of the response message is determined by the *<wsdai:ResponseFormat/>* element in the request message. This element contains the QName of a response message supported by the Data Service. Valid response messages are exposed by the Data Service using an informational property whose name takes the form:

```
<wsdai:RequestMessageResponseTypeList />
```

Where *RequestMessage* **MUST** be the name of one of the supported request message types contained in this list.

Realizations **MAY** choose to define interfaces containing statically typed response messages. When realizations define data access mechanisms that allow the consumer to define the response type the realization **MUST** provide an informational property of the form *RequestMessageResponseTypeList*.

6.2 Behavioral Properties

6.2.1 Readable

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

/wsdai:Readable

Has the value TRUE if a Data Service is able to return data in response to query operations. Otherwise has the value FALSE.

6.2.2 Writable

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

/wsdai:Writable

Has the value TRUE if a Data Service is able to update data represented by the Data Service in response to update or insert operations. Otherwise has the value FALSE.

6.2.3 ConcurrentAccess

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

/wsdai:ConcurrentAccess

Has the value TRUE if a Data Service is able to process more than one message concurrently. Otherwise it has the value FALSE.

6.2.4 TransactionInitiation

```
<xsd:element name="TransactionInitiation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="NotSupported"/>
      <xsd:enumeration value="Automatic"/>
      <xsd:enumeration value="Manual"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

/wsdai:TransactionInitiation

Describes under what circumstances a transaction is initiated in response to messages. Takes the values:

TransactionsNotSupported	Does not support Transactions
Automatic	Transaction initiated for each message
Manual	Transaction context under control of the Consumer

6.2.5 TransactionIsolation

```
<xsd:element name="TransactionIsolation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="NotSupported"/>
      <xsd:enumeration value="ReadUncommitted"/>
      <xsd:enumeration value="ReadCommitted"/>
      <xsd:enumeration value="RepeatableRead"/>
      <xsd:enumeration value="Serialisable"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

/wsdai:TransactionIsolation

Describes how transactions behave with respect to other ongoing transactions.

6.2.6 Sensitivity

```
<xsd:element name="Sensitivity">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="Insensitive"/>
      <xsd:enumeration value="Sensitive"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

/wsdai:Sensitivity

Describes the sensitivity to change of derived Data Service/Data Resource relationship when compared with the Data Service that created it. For example, when reading forwards and backwards in a RowSet, "Insensitive" means that you will read the same results when you read forwards and then backwards regardless of any changes in the Data Resource that created the RowSet. "Sensitive" means that any changes in the Data Resource that created the RowSet will be observed.

6.2.7 DataAccessTermsType

The properties defined in this section are collected together in an XML schema type that can be used as the base type for extension by the realizations.

```
<xsd:complexType name="DataAccessTermsType">
  <xsd:sequence>
    <xsd:element ref="wsdai:Readable" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:Writeable" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:ConcurrentAccess" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:TransactionInitiation" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element ref="wsdai:TransactionIsolation" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element ref="wsdai:Sensitivity" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

6.3 Messages

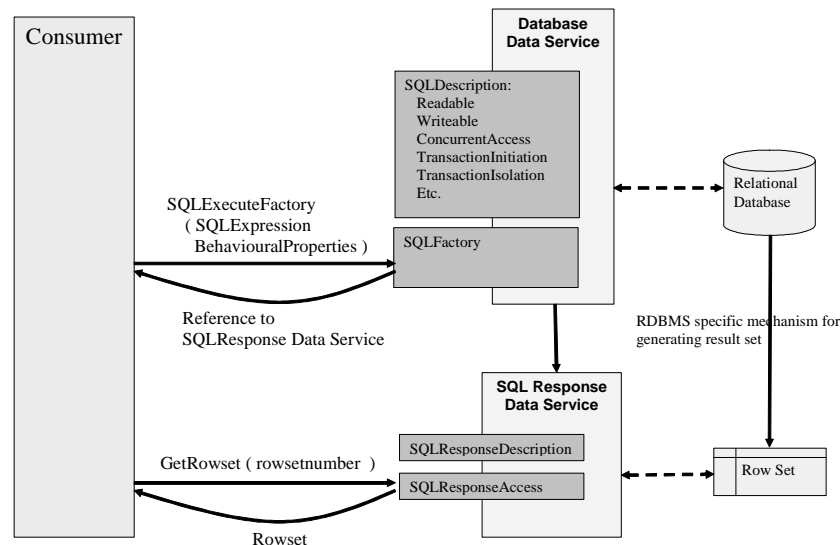
No data access messages are defined in the base specification. The specifications that extend this specification MUST define data access messages.

7. Data Factory

Messages defined within the Data Factory type of interface implement the factory pattern. Such messages create a new relationship between a Data Resource and a Data Service. In this way, a Data Service may be used to represent the results of a query or act as a place holder for data to be inserted into a Data Resource. A Data Factory has behavioral properties which dictate how a Data Service must behave on receiving factory messages.

The factory pattern MAY involve the creation of a new Data Resource. The factory pattern MAY involve the deployment of a web service. The details of how the factory pattern is applied are mapping dependent.

The factory pattern allows a new relationship between a Data Service and a Data Resource to result as the consequence of a messages exchange with a Data Service. This ability to derive one Data Service/Data Resource relationship from another or to provide derived views of the same Data Resources leads to a collection of notionally related Data Service/Data Resource relationships. For example:



The Database Data Service in this example presents a SQLFactor interface. The SQLExecuteFactory operation is used to construct the derived SQL Response Data Service. This service provides access to the RowSet resulting from an SQL expression against the Relational Database, assuming that the expression contains a SELECT statement. The RowSet could be stored as a table in a relational database or decoupled from the database, but the important distinction here is that the data is represented as a collection of rows that does not implement the SQLAccess portType. Instead the SQL Response Data Service presents the SQLResponseAccess collection of operations that allows the RowSet to be retrieved but does not provide facilities for submitting SQL expressions.

7.1 Message Patterns

The structure of a message implementing the factory pattern is:

```

<wsdai:RequestMessage>
  <wsdai:RequestDocument />
  <wsdai:ServiceTerms />
</wsdai:RequestMessage>

```

/wsdai:RequestMessage

This is the root element for a request message. The type of this element is specific to each message.

/wsdai:RequestMessage/RequestDocument

This is the request part of the message. The structure of this document is specific to the request message. This name of this element implies the language used by the request document.

/wsdai:RequestMessage/ServiceTerms

Service terms are the initial values of the behavioral properties of the Data Service that is to be constructed as a result of this message. They are specified by the Consumer. The type of this element is specific to the request message and, in particular, the type of Data Service that is expected to result from the processing of this message.

The DAIS working group has not adopted any particular grammar for advertising the initial values of a service's behavioral proprieties. Nor has it adopted a mechanism for negotiating these initial values. It is expected that these will be defined in other specifications, for example, WS Addressing [WS-Addressing]. It is the responsibility of the Consumer to provide initial values for all required behavioral properties given the constraints of the properties document schema.

Service terms are not universally applicable and will make sense only in the context of a Data Service implementing a particular interface. For example, it does not make sense to discuss the forward or backward nature of an iterator for a service that only accepts the SQLExecute message. The document containing service terms SHOULD also specify the type of interface that is required to result from the factory message. Typically this would be achieved by specifying the QName of the portType that is required in order to access the Data Resource.

A valid set of service term document schemas MAY be advertised using the *RequestMessageTermDocumentTypeList* informational property. If this is present the provided service term document MUST match one of the XML types included in this property.

The structure of the factory pattern response message is:

```
<wsdai:ResponseMessage>
  reference*
</wsdai:ResponseMessage>
```

/wsdai:ResponseMessage

This is the root element for a response message. The type of this element is specific to each message.

/wsdai:ResponseMessage /reference

The response to a message employing the factory pattern is zero or more references to the new Data Service/Data Resource relationships. The type of this reference is mapping specific.

7.2 Behavioral Properties

No behavioral properties are defined in the base specification.

7.3 Messages

No data factory messages are defined in the base specification. The specifications that extend this specification MAY define data factory messages.

8. Data Management

DAIS will not define further management interfaces until such time as it is clear what the scope of the work of the Web Services Distributed Management (WSDM) [WSDM] TC in OASIS is. It is hoped that the management interfaces that WSDM define will provide support and guidance for management of the behaviour of DAIS compliant Data Services.

8.1 Behavioral Properties

Data Management behavioural properties are not yet defined.

8.2 Messages

Data Management messages are not yet defined.

9. Mapping to WSDL

The Web Services Resource Framework (WSRF) is a set of proposed web services specifications that describe the WS-Resource construct as a means of expressing the relationship between stateful resources and web services. The reader is referred to the WSRF documentation for more information, for example “The WS-Resource Framework” [WS-Resource].

In mapping the DAIS properties and messages to WSRF, the DAIS working group assumes that a number of specifications are available for use:

- The specifications referenced by WS-I Basic Profile 1.0 [WS-I]
 - SOAP 1.1 [SOAP]
 - WSDL 1.1 [WSDL]
- WS-Addressing [WS-Addressing]
- WS-ResourceProperties [WS-ResourceProperties]
- WS-ResourceLifetime [WS-ResourceLifetime]

The use of WSDL 1.1 forces the manual aggregation of messages and properties from the DAIS base specification and from DAIS realizations into the port type for each Data Service. The interfaces described by the DAIS specifications and the realizations are not necessarily coherent on their own. They simply present sets of messages and properties that can be combined with other specifications from elsewhere into meaningful port types.

9.1 Data Services

In the context of this mapping a Data Service is web service that employs WSRF to allow access to Data Resources using the DAIS specified interfaces

In this mapping, WS-Addressing End Point References (EPR) implementing the implied resource pattern [WS-Resource] identifying the Data Service-Data Resource relationship. For example, a relational database may be accessed via a Data Service using a set of SQL access messages by referring to EPR1. Alternatively the same relational database may be accessed via a Data Service using a set of XML access messages by referring to EPR2.

Consumers may be provided with EPRs or may discover EPRs through registries. EPRs registered in registries will be associated with properties particular to a Data Service. These may include the Name and other descriptive information.

As in the previous example, many different EPRs can provide access to the same Data Resource through different interfaces. Many different EPRs can provide access to different Data Resources through the same Data Service.

To obtain a new EPR from an existing EPR, in order to access an existing data resource through a new interface, the Consumer must return to the registry or call a factory operation.

9.2 Data Resources

No assumption is made about the form that a Data Resource takes. The Data Service implements a set of messages and informational properties. The Data Service is able to interact with a Data Resource in order to process these messages and provide values for the informational properties.

9.3 Data Description

Informational properties are represented as resource properties [WS-ResourceProperties] and hence the structure of the properties for a given Data Service is statically defined in the service's WSDL document.

It is expected that resource properties will also be used to describe valid ranges of input values for certain message elements, in particular,

The Data Factory *RequestMessageTermDocumentTypeList*

A list of QNames that identifies the valid term structures that a factory operation can accept. Each of these structures combines a portType name with the term values that must be provided. This is used to tell the factory what type of service should be created and how it should be configured

The Data Access *RequestMessageResponseTypeList*

A list of QNames of valid response structures than a Data Access messages can generate.

In accordance with the approach outlined in the WS-ResourceProperties specification [WS-ResourceProperties], all property elements appropriate to a Data Service will be aggregated into a single resource property type which is in turn associated with the port type for the Data Service.

9.4 Data Access

The base specification defines no Data Access messages. Data Access messages defined in the realizations will appear in WSDL 1.1 definitions as operations on the port type of the Data Service. Due to the restrictions of WSDL 1.1, messages must be cut and pasted from the WSDL in the DAIS specification into the WSDL for the specific Data Service by the service developer.

Behavioral properties from the base specification and from any realization specifications are grouped together into an XML schema type which is particular to the Data Service being developed. An element of this type is exposed as the resource properties structure for the Data Service.

This grouping of behavioral properties also allows them to be passed easily with any factory messages.

9.5 Data Factory

The base specification defines no Data Factory messages. As with Data Access, Data Factory messages defined in the realizations will appear in WSDL 1.1 definitions as operations on the port type of the Data Service. Due to the restrictions of WSDL 1.1 messages must be cut and pasted from the WSDL in the DAIS specification into the WSDL for the Data Service by the service developer.

The base WSDL defines a term structure that provides a template, as described in Section 7.1, through restriction, for combining the structure containing the behavioral properties for a Data Service with the name of a port type to which they are relevant. This structure is used in the `<wsdai:ServiceTerms/>` element of messages implementing the factory pattern.

9.6 Data Management

Data Management messages have not yet been defined.

9.7 Faults

No faults are defined in the base specification.

9.8 Sessions

The creation of sessions across a Data Resource relies on further context information being provided in the messages, e.g., authentication information or other context information. Similarly, sessions across data resources also rely on further context information being provided.

9.9 Lifetime

The relationship between a Data Service and a Data Resource is constructed by messages implementing the factory pattern or by out of band means. The lifetime of the relationship is controlled using the properties and messages defined in the WS-ResourceLifetime specification.

10. Security Considerations

The Realizations of a Grid Data Service will use standard Grid security mechanisms as specified by OGSA Security working group combined with standard ways of relating Grid credentials and authorities to resource access rights. The assumption is that these standards will also indicate how to make information related to authentication, authorization, security, etc, available.

11. Conclusions

This document has described a proposal for a collection of top level interfaces for access to Data Resources as services, which are extended in companion documents to provide support for multiple data storage paradigms. The interfaces proposed are intended to be compatible with the architecture to be proposed by the GGF Open Grid Services Architecture working group. This is a work in progress, and feedback is welcomed on this document.

Editor Information

Mario Antonioletti,
EPCC,
University of Edinburgh,
James Clerk Maxwell Building,
Mayfield Road,
Edinburgh EH9 3JZ,
United Kingdom.

Malcolm Atkinson,
e-Science Institute,
15 South College Street,
Edinburgh EH8 9AA,
UK.

Amy Krause,
EPCC,
University of Edinburgh,
James Clerk Maxwell Building,
Mayfield Road,
Edinburgh EH9 3JZ,
United Kingdom.

Simon Laws,
IBM United Kingdom Limited,
Hursley Park,
Winchester,
Hampshire, SO21 2JN,

United Kingdom.

Susan Malaika,
IBM Corporation,
Silicon Valley Laboratory,
555 Bailey Avenue,
San Jose, CA 95141,
USA.

Norman W. Paton,
Department of Computer Science,
University of Manchester,
Oxford Road,
Manchester M13 9PL,
United Kingdom.

Dave Pearson,
Oracle Corporation Ltd,
Thames Valley Park,
Reading,
Berkshire RG6 1RA,
United Kingdom.

Greg Riccardi,
Department of Computer Science,
Florida State University,
Tallahassee, FL 32306-4530,
USA.

Contributors

Vijay Dialani, University of Southampton.
Allen Luniewski, IBM.
Inderpal Narang, IBM.
Steve Tuecke, Globus/ANL.
Jay Unger, IBM.

Acknowledgements

The DAIS Working Group of the Global Grid Forum is active, and many people have contributed to discussions within the group in recent months, including but not limited to: Bill Allcock, Dieter Gawlick, Shannon Hastings, Stephen Langella, Sastry Malladi, Paul Watson and Martin Westhead.

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

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

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

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

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

References

[DAISCIM]

S. Malaika and N.W. Paton, CIM Database Model for Data Access and Integration Services: Scenarios, CGS-WG Informational Draft, 10th Global Grid Forum, 24th February 2004.

[Data Services]

I. Foster, S. Tuecke, J. Unger, *OGSA Data Services*, DAIS-WG Informational Draft, 9th Global Grid Forum, 14th August 2003.

[Information Dissemination]

D. Gawlick, V. Gogate, C. Kantariiew, C. Madsen, S. Mishra, I. Narang, M. Subramanian, Information Dissemination in the Grid Environment, DAIS-WG Informational Draft, 10th Global Grid Forum, 19th September 2003.

[OGSA]

I. Foster (Ed), H. Kishimoto (Ed). *The Open Grid Services Architecture, Version 1.0*. Global Grid Forum.

[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.

[SOAP]

D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer. *Simple Object Access Protocol (SOAP) 1.1*. W3C Note 08 May 2000. Available from: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[WebRowSet]

J. Bruce, *Java Specification Request 114 JDBC Rowset Implementations*, Public Review 2, <http://jcp.org/en/jsr/detail?id=114>

[WS-Addressing]

A. Bosworth, D. Box (Ed), E. Christensen, F. Curbera (Ed), D. Ferguson, J. Frey, C. Kaler, D. Langworthy, F. Leymann, S. Lucco, S. Millet, N. Mukhi, M. Nottingham, D. Orchard, J. Shewchuk, T. Storey, S. Weerawarana. *Web Services Addressing (WS-Addressing)*. 13 March 2003. Available from: <ftp://www6.software.ibm.com/software/developer/library/ws-add200303.pdf>.

[WS-Agreement]

A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu. *Web Services Agreement Specification (WS-Agreement)*. <https://forge.gridforum.org/projects/graap-wg/document/draft-ggf-graap-agreement-0.9.pdf/en/1>

[WS-Context]

D. Bunting, M. Chapman, O. Hurley, M. Little, J. Mischinsky, E. Newcomer, J. Webber, K. Swenson. *Web Services Context (WS-Context)*, <http://www.oasis-open.org/committees/download.php/4344/WSCTX.pdf>

[WS-Coordination]

F. Cabrera, G. Copeland, T. Freund, J. Klein, D. Langworthy, D. Orchard, J. Schwchuk and T. Storey, *Web Services Coordination (WS-Coordination)*, <http://www-106.ibm.com/developerworks/library/ws-coor/>, 2002-a.

[WS-DAIR]

M. Antonioletti, B. Collins, A. Krause, S. Malaika, J. Magowan, S. Laws, N. W. Paton. *Web Services Relational Data Access and Integration*. DAIS-WG Informational Draft, 11th Global Grid Forum, 21st May 2004.

[WS-DAIX]

M. Antonioletti, A. Krause, S. Hastings, S. Langella, S. Malaika, S. Laws, N. W. Paton. *Web Service XML Data Access and Integration*. DAIS-WG Informational Draft, 11th Global Grid Forum, 21st May 2004.

[WSDM]

J. DeCarlo, I. Sedukhin (editors), *Web Services Distributed Management: Management Of Web Services*, Draft 0.5, 2 April 2004, <http://www.oasis-open.org/committees/download.php/6255/cd-wsdm-mows-0.5-20040402.pdf>

[WSDL]

E. Christensen, F. Curbera, G. Meredith and S. Weerawarana. *Web Services Description Language 1.1*. World Wide Web Consortium. W3C Note 15 March 2001. <http://www.w3.org/TR/wsdl>.

[WS-I]

K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, P. Yendluri. *Basic Profile 1.0*. 16 April 2004. Available from: <http://www.ws-i.org/Profiles/BasicProfile-1.0.html>.

[WS-Resource]

K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, *The WS-Resource Framework*, Version 1.0, May 3rd 2004, <http://www.oasis-open.org/committees/download.php/6796/ws-wsrf.pdf>

[WS-ResourceProperties]

S. Graham (Ed), K. Czajkowski, D.F. Ferguson, I. Foster, F. Leymann, T. Maguire, N. Nagaratnam, M. Nally, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, S. Weerawarana. *Web Services Resource Properties (WS-ResourceProperties)*. Available from; <http://www.ibm.com/developerworks/library/ws-resource/ws-resourceproperties.pdf>. Version 1.1 . 05 March 2004. ,

[WS-ResourceLifetime]

J. Frey (Ed), S. Graham (Ed), K. Czajkowski, D. F. Ferguson, I. Foster, F. Leymann, T. Maquire, N. Nagaratnam, M. Nally, T. Storey, I. Sedukhin, D. Snelling, S. Tuecke, W. Bambenepe, and S. Weerawarana. *Web Services Resource Lifetime (WS-ResourceLifetime)*. Version 1.1 05 March 2004. Available from: <http://www-106.ibm.com/developerworks/library/ws-resource/ws-resourcelifetime.pdf>.

[WS-Security]

OASIS Web Services Security 1.0 (WS-Security 2004) standard as of April 6th 2004, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[WS-Transaction]

F. Cabrera, G. Copeland, B. Fox, T. Freund, J. Klein, T. Storey and S. Thatte, *Web Services Transaction (WS-Transaction)*, <http://www.ibm.com/developerworks/library/ws-transpec/>, 2002-b.

[XQueryX]

Malhotra, J. Robie and M. Rys. *XML Syntax for XQuery 1.0 (XQueryX)*. W3C Working, See: <http://www.w3.org/TR/xqueryx>.

Appendix A – XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://www.ggf.org/namespaces/2004/05/WS-DAI"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdai="http://www.ggf.org/namespaces/2004/05/WS-DAI">

  <!-- general types -->

  <!-- the base type for input/output datasets -->
  <xsd:complexType name="DatasetType"/>
  <xsd:element name="Dataset" type="wsdai:DatasetType"/>

  <!-- the base type for query expressions -->
  <xsd:complexType name="ExpressionType"/>
  <xsd:element name="Expression" type="wsdai:ExpressionType"/>

  <!-- the base types for terms passed into operations following -->
  <!-- the factory pattern -->
  <xsd:complexType name="TermDocumentType">
    <xsd:sequence>
      <xsd:element name="PortType" type="xsd:QName" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element name="Terms" minOccurs="1" maxOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:any/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="TermDocument" type="wsdai:TermDocumentType" abstract="true" />

  <!-- presents a list of response types. The consumer selects from -->
  <!-- this list when constructing the response format element of a -->
  <!-- message for a direct access message -->
  <xsd:complexType name="ResponseTypeListType">
    <xsd:sequence>
      <xsd:element name="ResponseTypeQName" type="xsd:QName" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

```
        </xsd:sequence>
    </xsd:complexType>

    <!-- the element used in direct access messages to pass the QName -->
    <!-- of the require response format -->
    <xsd:element name="ResponseFormat" type="xsd:QName"/>

    <!-- Presents a list of term document names that can validly be passed -->
    <!-- into an operation following the factory pattern -->
    <xsd:complexType name="TermDocumentTypeListType">
        <xsd:sequence>
            <xsd:element name="TermDocumentTypeQName" type="xsd:QName" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>

<!-- data description -->

    <xsd:element name="Name" >
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:any minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Description">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:any minOccurs="0" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

<!-- data access -->

    <xsd:element name="Readable" type="xsd:boolean" />

    <xsd:element name="Writeable" type="xsd:boolean" />

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



```
<xsd:element name="TransactionInitiation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="NotSupported"/>
      <xsd:enumeration value="Automatic"/>
      <xsd:enumeration value="Manual"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="TransactionIsolation">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="NotSupported"/>
      <xsd:enumeration value="ReadUncommitted"/>
      <xsd:enumeration value="ReadCommitted"/>
      <xsd:enumeration value="RepeatableRead"/>
      <xsd:enumeration value="Serialisable"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:element name="Sensitivity">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="Insensitive"/>
      <xsd:enumeration value="Sensitive"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<xsd:complexType name="DataAccessTermsType">
  <xsd:sequence>
    <xsd:element ref="wsdai:Readable" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:Writeable" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:ConcurrentAccess" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:TransactionInitiation" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:TransactionIsolation" minOccurs="0" maxOccurs="1"/>
    <xsd:element ref="wsdai:Sensitivity" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

```

<xsd:element name="DataAccessTerms" type="wsdai:DataAccessTermsType"/>

<!-- data factory -->

<!-- data management -->
<!-- TODO - how should we express these rules - not an enumeration! -->
<xsd:element name="DestructionRule">
  <xsd:simpleType>
    <xsd:restriction base="xsd:token">
      <xsd:enumeration value="ResourceLifetime"/>
      <xsd:enumeration value="Inactivity"/>
      <xsd:enumeration value="ParentResource"/>
      <xsd:enumeration value="DataResource"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

</xsd:schema>

```

Appendix B – WSDL

```

<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions name="wsdai"
  targetNamespace="http://www.ggf.org/namespaces/2004/05/WS-DAI"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdai="http://www.ggf.org/namespaces/2004/05/WS-DAI">

  <!-- WSDL IMPORTS ##### -->

  <!-- WSDL TYPES ##### -->
  <wsdl:types>
    <!-- ##### -->
    <!-- ### DAIS Base Schema ### -->
    <!-- ##### -->
    <xsd:schema targetNamespace="http://www.ggf.org/namespaces/2004/05/WS-DAI" elementFormDefault="qualified">

      <!-- ##### -->
      <!-- ### DataService Schemas ### -->
    </xsd:schema>
  </wsdl:types>

```

```
<!-- ##### -->
<xsd:element name="DataServiceDescription">
  <xsd:complexType>
    <xsd:sequence>
      <!-- from wsdaï - data description - properties of the data resource -->
      <xsd:element ref="wsdaï:Name" minOccurs="0" maxOccurs="1" />
      <xsd:element ref="wsdaï:Description" minOccurs="0" maxOccurs="1"/>

      <!-- from wsdaï - data access - properties controlling access -->
      <!-- behaviour -->
      <xsd:element ref="wsdaï:DataAccessTerms" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>

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

  <!-- none defined -->

</wsdl:definitions>
```