GWD-R
GGF DAIS Working Group

Editors
Mario Antonioletti, University of Edinburgh
Brian Collins, IBM
Amy Krause, University of Edinburgh
Simon Laws, IBM
James Magowan, IBM
Susan Malaika, IBM
Norman W Paton, University of Manchester

Category: INFORMATIONAL

17 June, 2005

Web Services Data Access and Integration – The Relational Realisation (WS-DAIR)

Status of This Memo

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

Copyright Notice

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 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, *Web Services Data Access and Integration: The Relational Realisation (WS-DAIR)*, presents a specification for a collection of data access interfaces for relational data resources, which extends interfaces defined in the *Web Services Data Access and Integration* document [WS-DAI].

Related DAIS specifications define how other 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

1. Introduction

Data access plays a central role for many types of Grid applications. Data access generally involves the retrieval, manipulation and insertion of data, which may be stored using a range of different formats and infrastructures.

This document presents a specification for a collection of data access interfaces for relational data resources. A relational data resource is a data source/sink that is characteristic of relational database systems, e.g., can be queried or updated using SQL or any other suitable relational query/update language. The interfaces are categorized according to the support they provide for:

- Data description: the provision of metadata about the pertinent characteristics of a data resource that a service may wish to expose as well as associated properties that affect the interaction between a service and the data resource.
- Data access: the provision of access to data through a service interface.
- Data factory: the provision of indirect access to data resources through new service interfaces.

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

1.1 Specification Scope

The data access and data factory interface categories as well as the role of data description properties in the provision of service-based access to data resources are described in the *Web Services Data Access and Integration* specification [WS-DAI]. This specification extends those interfaces and properties to allow access to and provide descriptions of relational data resources. The relational data resources are assumed to be composed of tabular data structures such as relations and result setswhich are typically accessed either using SQL queries or by row iteration, respectively.

1.2 Specification Organisation

This specification separates the function of a data service from its operational representation as expressed in WSDL.

The relational model is described using the terminology defined in Section 3 and employs the concepts described in Section 4. Sections 5, 6 and 7 describe the interfaces for posing SQL queries, accessing the results of SQL queries and for iterating through result sets respectively.

A mapping of the relational model to the Web Services Resource Framework (WSRF) [WS-Resource] is described in Section 8, Section 9 discusses security and Section 10 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.

Here a data service provides SQLAccess, SQLResponseAccess and SQLRowSetAccess interfaces for a relational data service that is associated with a relational database.

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, 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/>).

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: “*” indicates zero or more elements, “+” indicates one or more elements and “?” indicates zero or one element only. Where no notation is associated with an element only one instance of the element is expected.

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

Prefix	Namespace
xsd	http://www.w3.org/2001/XMLSchema
wsdai	http://www.ggf.org/namespaces/2005/06/WS-DAI
wsdair	http://www.ggf.org/namespaces/2005/06/WS-DAIR
wrs	http://java.sun.com/xml/ns/jdbc

3. Terminology

The model independent terminology, i.e., data resource, data service, consumer and data set, is given in the *Web Services Data Access and Integration* document [WS-DAI], henceforth referred to as the WS-DAI document.

3.1 Relational Data Resource

A relational data resource is a data source/sink that is characteristic of relational database systems, e.g., can be queried or updated using SQL or any other suitable relational query/update language.

4. Concepts

4.1 Interfaces

DAIS classifies its interfaces into three broad categories, which are defined in the WS-DAI specification and extended in this document for relational data resources. Note that the word interface refers to the collections of messages and XML structures that describe the ways in which a consumer can validly, through DAIS, 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.

4.1.1 Data Description

The data description provides a set of properties allowing a description of data represented by data services or of the capabilities of the underlying relational data resource to be made. The model-independent specification for these is given in the WS-DAI document. Here they are extended to provide a description of relational data resources. The main points of extension for relational data resources are:

- *SQLAccessDescription*: extends the properties enumerated in the WS-DAI document to provide information about access properties and capabilities of the underlying relational data resource.
- *SQLResponseDescription*: extends the properties enumerated in the WS-DAI document to provide information about data returned or the result of an interaction with a relational data resource.
- *SQLRowSetDescription*: provides information about a particular instance of a query result that a data service may represent. This interface will make available information about the schema for representing the query result and the number of rows within the SQLRowSet.

These capabilities are described in Sections 5 to 7.

4.1.2 Data Access

Data access operations allow relational data resources to be modified through insertion, updates or deletes, or queried through an appropriate language. When a relational data resource also supports XML access capabilities then the interfaces described in the WS-DAIX specification may be used in conjunction with the relational interfaces to provide access data stored in this format.The following data access interfaces are defined in this specification:

- *SQLAccess*: provides access to a relational data resource.
- *SQLResponseAccess*: provides access to each type of Response that can result from the execution of a SQLExpression.
- *SQLRowSetAccess*: provides access to a set of rows, which are usually the result of a SQLExpression containing a SELECT statement.

These are covered in more detail in Sections 5 to 7.

4.1.3 Data Factory

The data factory interfaces allow data represented in relational data resources, usually as the result of a query, to be instantiated as a data service. The specialization in this instance deals with the type of SQLExpression that can be passed to a *Data Factory* to expose the results in a meaningful fashion. The properties and interfaces that will be supported by such a data service are specified in the schema for the creation parameters. *Data Factory* specializations defined in this specification are:

- *SQLAccessFactory*: provides access to a relational data resource.
- *SQLResponseFactory*: provides access to a SQL response.

These are covered in more detail in Sections 5 to 7.

4.2 Relationships with other specifications

WS-DAIR does not specify its own query/update languages for relational data resources. Instead, it acts as a channel for existing relational query and update languages to be conveyed to the appropriate data resources, in this instance relational data resources or a data resource that supports relational type queries. As such WS-DAIR relies on existing relational query and update languages. In this document, interface support is provided for languages based on the following standards:

- SQL: an ISO standard defining a language for querying and updating relational data resources [SQL].

- WebRowSet: a Java Community Process standard for relational results is one of the valid *ResponseFormats* for responses from SQLAccess operations [JSR114].
- CIM: is the Common Information Model, a DMTF standard, to which the DAIS-WG has submitted a proposal for extension to include relational database properties [CIM].

5. SQL

5.1 SQLAccessDescription

The elements described here are contained within a SQLPropertyDocument element.

5.1.1 LanguageCapabilities

This element allows the service provider to enumerate the different language capabilities that are supported by the underlying data resource.

```
<xsd:simpleType name=”LanguageCapabilitiesType”>
    <xsd:union>
        <xsd:simpleType>
            <xsd:restriction base="xsd:token">
                <xsd:enumeration value="SQL92Expression"/>
                <xsd:enumeration value="SQL99Expression"/>
                <xsd:enumeration value="SQL03Expression"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType>
            <xsd:restriction base="xsd:token"/>
        </xsd:simpleType>
    </xsd:union>
</xsd:simpleType>

<xsd:element name="LanguageCapabilities"
             type="wsdair:LanguageCapabilitiesType" />
```

/wsdair:LanguageCapabilities

Describes the dialect of the SQL languages that the underlying relational database management system should support.

5.1.2 CIM Description

This element is a place holder that will allow an XML representation of the DMTF CIM to be added providing metadata about the underlying relational data resource. Neither the XML representation of the CIM model, nor the name space, has been specified yet.

```
<xsd:element name="CIMDescription">
 <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="DMTF-CIM-Namespace-TBD"
               processContents="strict"
               minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
 </xsd:complexType>
</xsd:element>
```

/wsdair:CIMDescription

Allows content from the XML DMTF CIM to be exposed through the data service.

5.2 SQLAccess

This *SQLAccess* interface provides access to the underlying relational data resource by means of SQL statements.

5.2.1 Overview

Data Access collects together messages that directly access and modify the data represented by a data service along with the behavioral properties that describe the behavior of these access messages, as, for example, illustrated in Figure 1.

A relational data service implements the SQLAccess operations and exposes the *SQLAccessDescription* properties. In this example a consumer uses the *SQLExecute* message to submit a *SQLExpression*. The associated *SQLExecuteResponse* message will contain some combination of *SQLExecuteResponseTypeList*. The actual combination will depend upon the actual *SQLExpression,* for example:

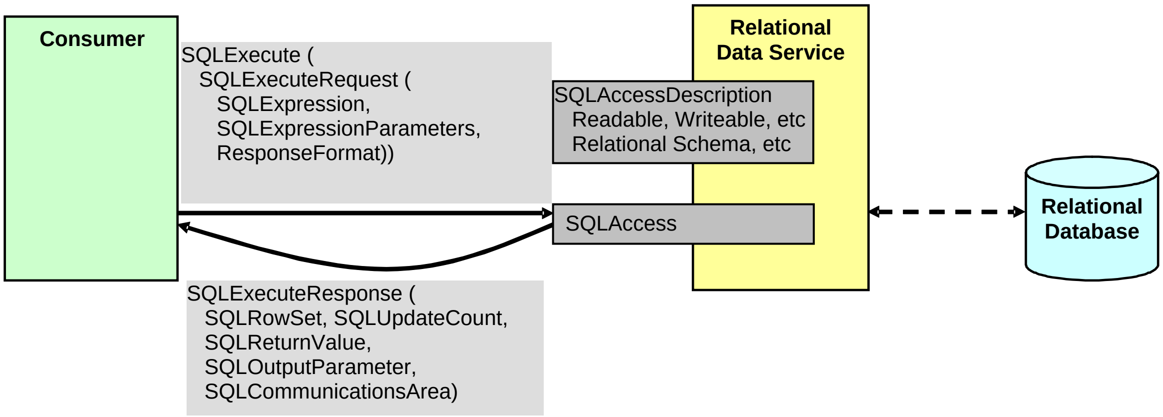

Figure 1 – Overview – SQLAccess

- Usage of SELECT produces an SQLExecuteResponse with:
  - o *(SQLRowSet*, *SQLCommunicationsArea**)
- Usage of INSERT, UPDATE, DELETE produces an SQLExecuteResponse with:
  - o *(SQLUpdateCount, SQLCommunicationsArea**)
- Usage of a StoredProcedure produces an SQLExecuteResponse with:
  - o *(SQLRowSet**,*SQLReturnValue**,*SQLOutputParameter**,*SQLCommunicationsArea**)
- Usage of a UserDefinedFunction produces an SQLExecuteResponse with:
  - o *(SQLReturnValue*+*, SQLCommunicationsArea**)

The Consumer will need to process the *SQLExecuteResponse* appropriately.

5.2.2 Operations

5.2.2.1 SQLAccess::GetSQLPropertyDocument

Provides a mechanism for obtaining a copy of the SQLAccessDescription to be obtained.

Input

- *GetSQLDocumentPropertyRequest*
  - o *ResourceName* – optional name of the resource for which properties are required.

Output

- *GetSQLDocumentPropertyResponse*
    - *PropertyDocument* – the properties described in the data description section.

Faults

- *InvalidResourceName* – the supplied resource name is not known to the service.

5.2.2.2 SQLAccess::SQLExecute

Directs a SQL*Expression* and optional *SQLExecuteRequestParameters* to a relational data resource.

The *SQLExecuteRequestParameters* are primarily for use with Stored Procedures and User Defined Functions. However, it is also intended that a SQL*ExecuteRequestParameter* be used to provide a dataset reference to a *SQLExpression* containing a bulk load or similar update statement.

Input

- *SQLExecuteRequest* – the SQLExecute operation that is to be run on the relational data resource.
    - *ResourceName* – optional name of the resource the messages should be targeted at.
    - *SQLExpression* – any SQL statement.
    - *SQLExpressionParameters* – any *SQL parameters that need to be substituted into* the supplied SQLExpression if it contains any SQL parameter markers.
    - *ResponseFormat* – the format(s), selected from the *SQLExecuteResponseTypeList* property, which the *SQLExecuteResponse* will conform to.

Output

- *SQLExecuteResponse* – the *SQLExecuteResponse* returned in the *ResponseFormat*s from the SQLExecute operation.
    - *SQLRowSet* – e.g. WebRowSet see [JSR114].
    - *SQLUpdateCount* – the number of rows that were affected by an SQL update if this was the type of SQL statement used.
    - *SQLReturnValue* – the return value from any stored procedure.
    - *SQLOutputParameter* – where any output from a SQL stored procedure output parameter goes*.*
    - *SQLCommunicationsArea* – exposes any output from the SQL Communications Area.
        - *SQLState* – an XOPEN or SQL99 code identifying the Exception, Warning or Message.
        - *VendorCode* – a database vendor-specific code for the Exception, Warning or Message.
        - *MessageText* – a text description of the Exception, Warning or Message.

Faults

- *InvalidSQLExecuteRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLExpressionParameters* – Parameters do not match SQLExpression.
- *InvalidResponseFormat* – ResponseFormat not valid.
- *OtherFault* – any other fault.

5.3 SQLAccessFactory

The SQLExecuteFactory operation is used to create a service representing a relational data resource, which fulfills the desired behavior, exposes the desired interfaces and represents the results of the SQL query.

5.3.1 Overview

This factory pattern allows a data service to relational data resource relationship to be established as a result of messages going to another data service. This ability to derive one data service from another to provide different views of relational data resources leads to a collection of notionally related data service instances, for example, see Figure 2.

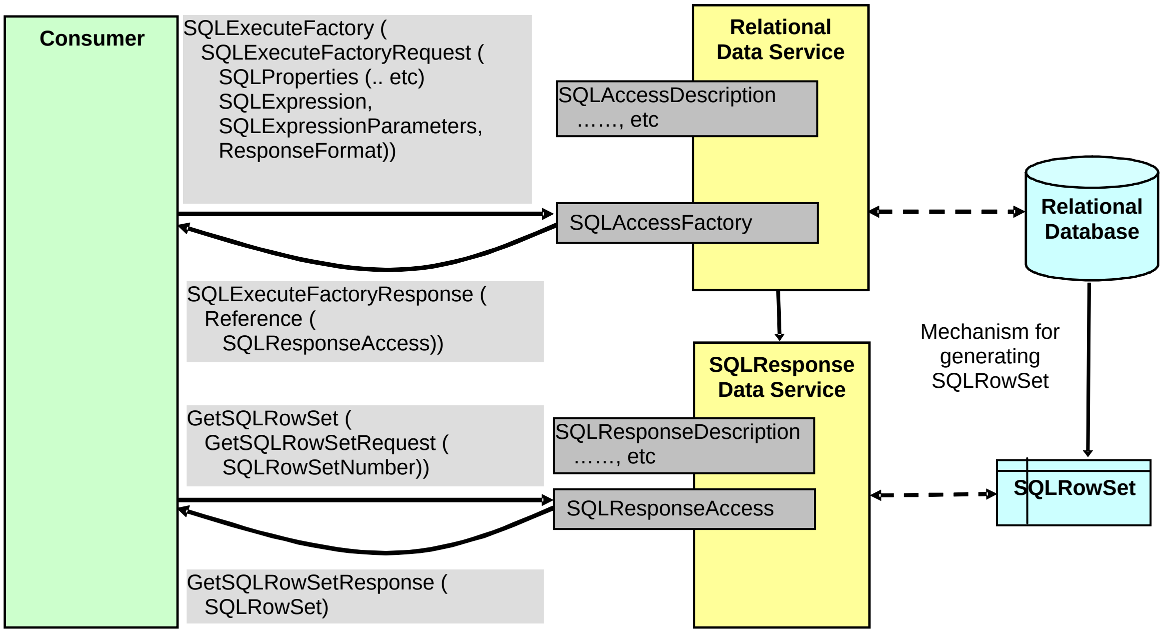

Figure 2 – Overview – SQLAccessFactory

The example in Figure 2 presents a SQLAccessFactory interface. The SQLExecuteFactory operation is used to construct the derived SQLResponse data service. This service provides access to the *SQLRowSet* resulting from a *SQLExpression* against the Relational Database, assuming that the expression contains a SELECT statement. The *SQLRowSet* is a subset or restriction of the data in the database and is presented in tabular form. The *SQLRowSet* 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 SQLResponse data service presents the *SQLResponseAccess* collection of operations that allows the *SQLRowSet* to be retrieved but does not provide facilities for submitting SQL expressions.

5.3.2 Operations

5.3.2.1 SQLAccessFactory::SQLExecuteFactory

Create a new data service that corresponds to the results of a SQL Query.

Input

- *SQLExecuteFactoryRequest*
  - *ResourceName* – optional name of the resource the messages should be targeted at.
  - *PropertiesDocument* – DataAccessPropertyType for target data service.
  - *SQLExpression* – any SQL statement.
  - *SQLExpressionParameters* – any *SQL parameters that need to be substituted into* the supplied SQLExpression if it contains any SQL parameter markers.

Output

- *SQLExecuteFactoryResponse*

  - *Reference* – to SQLResponseAccess operation.

Faults

- *InvalidSQLExecuteFactoryRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLBehavioralProperties* – Properties not valid.
- *InvalidSQLExpressionParameters* – SQLExpressionParameters do not match SQLExpression.
- *InvalidResponseFormat* – ResponseFormat not valid.
- *OtherFault* – any other fault.

6. SQLResponse

6.1 SQLResponseDescription

6.1.1 SQLResponseItem

```
<xsd:complexType name="SQLResponseItemType">
  <xsd:sequence>
    <xsd:element name="SQLResponseItemSequenceNumber" type="xsd:int" />
    <xsd:element name="SQLResponseItemFormatType" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="SQLResponseItem"
             type="wsdair:SQLResponseItemType"/>
```

/wsdaisr: SQLResponseItem

This comprises two properties, namely *SQLResponseItemSequenceNumber,* the Sequence Number of a SQLResponse Item in the *SQLExecuteResponse* which is paired with *SQLResponseItemFormatType,* the Format Type for that same SQLResponse Item.

6.1.2 NumberOfSQLRowSets

```
<xsd:element name="NumberOfSQLRowSets" type="xsd:int" />
```

/wsdaisr:NumberOfSQLRowSets

The total number of *SQLRowSets* in the *SQLExecuteResponse.*

6.1.3 NumberOfSQLUpdateCounts

```
<xsd:element name="NumberOfSQLUpdateCounts" type="xsd:int" />
```

/wsdaisr:NumberOfSQLUpdateCounts

The total number of *SQLUpdateCounts* in the *SQLExecuteResponse.*

6.1.4 NumberOfSQLReturnValues

```
<xsd:element name="NumberOfSQLReturnValues" type="xsd:int" />
```

/wsdaisr:NumberOfSQLReturnValues

The total number of *SQLReturnValues* in the *SQLExecuteResponse.*

6.1.5 NumberOfSQLOutputParameters

```
<xsd:element name="NumberOfSQLOutputParameters" type="xsd:int" />
```

/wsdaisr:NumberOfSQLOutputParameters

The total number of *SQLOutputParameters* in the *SQLExecuteResponse.*

6.1.6 NumberOfSQLCommunicationsAreas

```
<xsd:element name="NumberOfSQLCommunicationsAreas" type="xsd:int" />
```

/wsdaisr:NumberOfSQLCommunicationsAreas

The total number of *SQLCommunicationsAreas* in the *SQLExecuteResponse.*

6.2 SQLResponseAccess

This allows access to each *SQLExecuteResponseType* in the *SQLExecuteResponse* data by executing the appropriate SQLResponseAccess operation.

6.2.1 Operations

6.2.1.1 SQLResponseAccess::GetSQLResponsePropertyDocument

Allows a copy of the SQLResponseDescription document to be obtained.

Input

- *GetSQLResponseDocumentPropertyRequest*
  - *ResourceName* – optional name of the resource from which the data is to be obtained.

Output

- *GetSQLResponseDocumentPropertyResponse*
  - *PropertyDocument* – the properties described in the data description section.

Faults

- *InvalidResourceName* – the supplied resource name is not known to the service.

6.2.1.2 SQLResponseAccess::GetSQLResponseItem

Return a specified number of Items from a service that represents a SQL Response. This provides an alternative way to access SQL Response Items to the operations (GetSQLRowSet, GetSQLUpdateCount, etc) for SQLResponseAccess. The Response Format for each Item is obtained from the associated SQLResponseItem property.

Input

- *GetSQLResponseItemRequest*
  - *ResourceName* – optional name of the resource from which the data is to be obtained.
  - *StartPosition* – the position of the first SQL Response Item to be returned. (First Item is position 1).
  - *Count* – the number of SQL Response Items.

Output

- *GetSQLResponseItemResponse*
  - *SQLResponseItem* – e.g. SQLRowSet, SQLUpdateCount, etc.

Faults

- *InvalidGetSQLResponseItemRequest* – XML syntax error or XML schema non-compliance.
- *InvalidStartPosition* – not a valid StartPosition; cannot start with SQL Response Item specified (out of bounds value).
- *InvalidCount* – not a valid Count; cannot return that number of Response Items.

- *OtherFault* – any other fault.

6.2.1.3 SQLResponseAccess::GetSQLRowSet

Get a *SQLRowSet* from the *GetSQLRowSetResponse*.

Input

- *GetSQLRowSetRequest*
    - *ResourceName* – optional name of the resource from which the data is to be obtained.
    - *SQLRowSetNumber* – the number of the required *SQLRowSet.*

Output

- *GetSQLRowSetResponse*
    - *SQLRowSet* – the requested *SQLRowSet* e.g. WebRowSet see [JSR114].

Faults

- *InvalidGetSQLRowSetRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLRowSetNumber* – not a valid *SQLRowSetNumber.*
- *OtherFault* – any other fault.

6.2.1.4 SQLResponseAccess::GetSQLUpdateCount

Get a SQL*UpdateCount* from the *GetSQLUpdateCountResponse*.

Input

- *GetSQLUpdateCountRequest*
    - *ResourceName* – optional name of the resource from which the data is to be obtained.
    - *SQLUpdateCountNumber* – the number of the required *SQLUpdateCount.*

Output

- *GetSQLUpdateCountResponse*
    - *SQLUpdateCount* – the requested *SQLUpdateCount.*

Faults

- *InvalidSQLUpdateCountRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLUpdateCountNumber* – not a valid *SQLUpdateCountNumber.*
- *OtherFault* – any other fault.

6.2.1.5 SQLResponseAccess::GetSQLReturnValue

Get a *SQLReturnValue* from the *GetSQLReturnValueResponse*.

Input

- *GetSQLReturnValueRequest*
    - *ResourceName* – optional name of the resource from which the data is to be obtained.
    - *SQLReturnValueNumber* – the number of the required *SQLReturnValue.*

Output

- *GetSQLReturnValueResponse*
    - *SQLReturnValue* – the requested *SQLReturnValue.*

Faults

- *InvalidGetSQLReturnValueRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLReturnValueNumber* – not a valid *SQLReturnValueNumber.*
- *OtherFault* – any other fault.

6.2.1.6 SQLResponseAccess::GetSQLOutputParameter

Get a SQL*OutputParameter* from the *GetSQLOutputParameterResponse*.

Input

- *GetSQLOutputParameterRequest*
    - *ResourceName* – optional name of the resource from which the data is to be obtained.
    - *SQLOutputParameterNumber* – the number of the required *SQLOutputParameter.*

Output

- *GetSQLOutputParameterResponse*
    - *SQLOutputParameter – the requested SQLOutputParameter.*

Faults

- *InvalidSQLOutputParameterRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLOutputParameterNumber* – not a valid *SQLOutputParameterNumber.*
- *OtherFault* – any other fault.

6.2.1.7 SQLResponseAccess::GetSQLCommunicationsArea

Get a *SQLCommunicationsArea* from the *GetSQLCommunicationsAreaResponse*.

Input

- *GetSQLCommunicationsAreaRequest*
    - *ResourceName* – optional name of the resource from which the data is to be obtained.
    - *SQLCommunicationsAreaNumber* – the number of the required *SQLCommunicationsArea.*

Output

- *GetSQLCommunicationsAreaResponse*
    - *SQLCommunicationsArea – the requested SQLCommunicationsArea.*
        - *SQLState* – an XOPEN or SQL99 code identifying the Exception, Warning or Message.
        - *VendorCode* – a database vendor-specific code for the Exception, Warning or Message.
        - *MessageText* – a text description of the Exception, Warning or Message.

Faults

- *InvalidSQLCommunicationsAreaRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLCommunicationsAreaNumber* – not a valid *SQLCommunicationsAreaNumber.*
- *OtherFault* – any other fault.

6.3 SQLResponseFactory

The SQLResponseFactory is used to provide access to a row set as a data resource, by way of a data service.

6.3.1 Overview

The example in Figure 3 presents a SQLResponseFactory interface. The SQLExecuteFactory operation is used to construct the derived SQLResponse data service, the SQLResponseFactory operation of which is in turn used to construct the derived SQLRowSet data service. This service provides access to tuples in the *SQLRowSet* resulting from a *SQLExpression* against the Relational Database.

6.3.2 Operations

6.3.2.1 SQLResponseFactory::SQLRowSetSelectionFactory

Get a Refer*ence* to a *SQLRowSetAccess* from the *SQLRowSetSelectionFactoryResponse*.

Input

- *SQLRowSetSelectionFactoryRequest*

- o *PropertiesDocument* – DataAccessPropertyType for target data service.
- o *SQLRowSetSelectionNumber* – the number of the required *SQLRowSet.*

Output

- *SQLRowSetSelectionFactoryResponse*
  - o *Reference – to SQLRowSetAccess* operation which provides access to the requested SQLRowSet.

Faults

- *InvalidSQLRowSetSelectionFactoryRequest* – XML syntax error or XML schema non-compliance.
- *InvalidSQLRowSetSelectionNumber* – not a valid *SQLRowSetSelectionNumber.*
- *OtherFault* – any other fault.

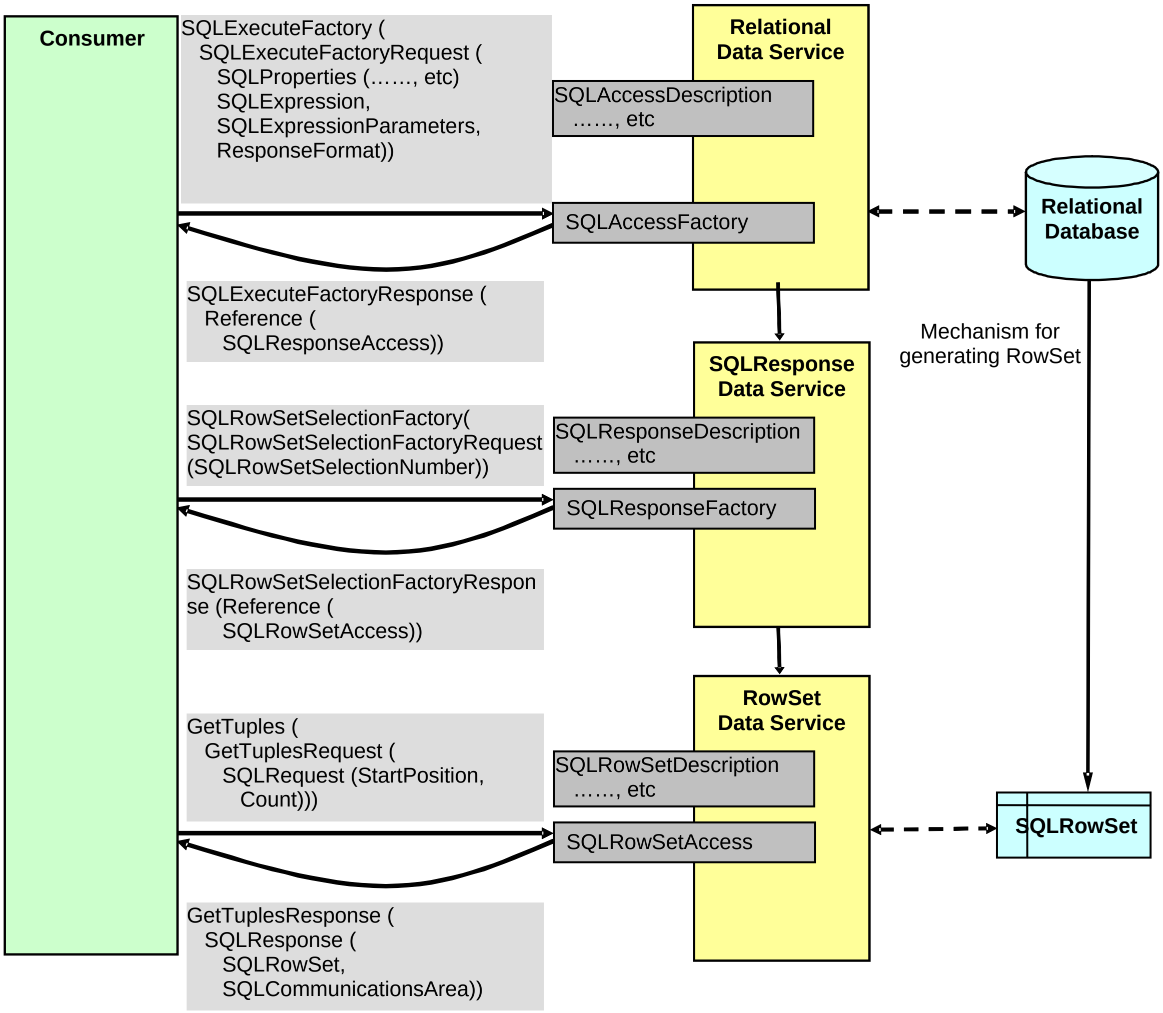

Figure 3 – Overview – SQLResponseFactory

7. SQLRowSet

7.1 SQLRowSet Description

7.1.1 AccessMode

```
<xsd:complexType name="AccessModeType">
  <xsd:sequence>
    <xsd:element name="Content" type="xsd:string"  minOccurs="0"
```

```
                 maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="AccessMode" type="wsdair:AccessModeType" />
```

/wsdairs:AccessMode

Describes how the SQLRowSet can be navigated, for example sequentially in a forward direction; whether random access is allowed, etc. Possible values would be one or more from a set of values which will be implementation specific, for example ForwardSequential, RandomAccess. The "Content" element at the moment is purely acting as a temporary placeholder.

7.1.2 CursorHeldOverTxnBoundary

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

/wsdairs:CursorHeldOverTxnBoundary

Describes whether the SQLRowSet can still be navigated after a transaction has been committed.

7.1.3 RowSchema

```
<xsd:complexType name="RowSchemaType">
  <xsd:sequence>
    <xsd:element ref="wrs:metadata"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="RowSchema" type="wsdairs:RowSchemaType" />
```

/wsdairs:RowSchema

For example, WebRowSet, see [JSR114].

7.1.4 NumberOfRows

```
<xsd:element name="NumberOfRows" type="xsd:int" />
```

/wsdairs:NumberOfRows

The total number of rows in the result set.

7.2 SQLRowSetAccess

This allows access to the underlying data by means of rows.

7.2.1 Operations

7.2.1.1 SQLRowSetAccess::GetTuples

Return a specified number of tuples from a service that represents a result set.

Input
- *GetTuplesRequest*
  - o *ResourceName* – optional name of the resource from which the data is to be obtained.
  - o *StartPosition* – the position of the first tuple to be returned (First tuple is position 1).
  - o *Count* – the number of tuples.

Output
- *GetTuplesResponse*
  - o *SQLRowSet* – e.g. WebRowSet see [JSR114].

  - *SQLCommunicationsArea* –
    - *SQLState* – an XOPEN or SQL99 code identifying the Exception, Warning or Message.
    - *VendorCode* – a database vendor-specific code for the Exception, Warning or Message.
    - *MessageText* – a text description of the Exception, Warning or Message.

Faults

- *InvalidGetTuplesRequest* – XML syntax error or XML schema non-compliance.
- *InvalidStartPosition* – not a valid StartPosition; cannot start with tuple specified (out of bounds value).
- *InvalidCount* – not a valid Count; cannot return that number of tuples.
- *OtherFault* – any other fault.

8. Mapping to WSRF

For a mapping to the Web Services Resource Framework (WSRF) [WS-Resource] proposal see the following Sections:

- SQLAccess
  - WSDL Interfaces – Appendix A.1.
  - XML Schema – Appendix A.2.
  - WSDL – Appendix A.3.
- SQLResponseAccess
  - WSDL Interfaces – Appendix B.1.
  - XML Schema – Appendix B.2.
  - WSDL – Appendix B.3.
- SQLRowSetAccess
  - WSDL Interfaces – Appendix C.1.
  - XML Schema – Appendix C.2.
  - WSDL – Appendix C.3.

9. Security Considerations

The relational realization of a Grid data service will use standard grid security mechanisms as specified by the AUTHZ Working Group[1] 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.

10. Conclusion

This document has discussed a specialization of the interfaces defined in the *Web Services Data Access and Integration* document [WS-DAI] and the additional capabilities required to properly address relational data resources.

[1] See http://forge.gridforum.org/projects/authz-wg.

Editor Information

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

Brian M Collins
11 St Stephen Road,
Winchester,
SO22 6DE,
United Kingdom.
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.

James Magowan,
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,
School of Computer Science,
University of Manchester,
Oxford Road,
Manchester M13 9PL,
United Kingdom.

Contributor Information

Vijay Dialani, IBM.
Greg Riccardi, Florida State University.
Shannon Hastings, Ohio State University.

Stephen Langella, Ohio State University.

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, Allen Luniewski, Sastry Malladi, Inderpal Narang, Steve Tuecke, Jay Unger, Paul Watson, Martin Westhead, Patrick Dantressangle.

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

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

[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.

[SQL]

Information technology -- Database languages -- SQL -- Part 14: XML-Related Specifications (SQL/XML), ISO/IEC 9075-14:2003, http://www.iso.ch/iso/en/stdsdevelopment/tc/tclist/TechnicalCommitteeStandardsListPage.TechnicalCommitteeStandardsList?COMMID=160&printable=true.

There are other earlier versions of the SQL specifications, e.g., SQL99

[JSR114]

J. Bruce, JSR-000114 JDBC RowSet Implementations, Final Release, 07 April 2004. http://jcp.org/aboutJava/communityprocess/final/jsr114/index.html.

[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://forge.gridforum.org/projects/ggf-editor/document/GFD.30/en/1.

[WS-DAI]

M. Antonioletti, M. Atkinson, S. Laws, S. Malaika, N. W. Paton D. Pearson and G. Riccardi. *Web Services Data Access and Integration (WS-DAI)*. DAIS-WG Informational Draft, 14th Global Grid Forum, 17th June, 2005.

[WS-Resource]

S. Graham (Ed), A. Karmarkar (Ed), J. Mischkinsky (Ed), I. Robinson (Ed), I. Sedukhin (Ed). Web Services Resource 1.2 (WS-Resource), Version 1.2, Committee Draft 01, 19 May 2005. http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-cd-01.pdf

[CIM]

DMTF Common Information Model Standards
http://www.dmtf.org/standards/cim/.

Appendix A.1 – SQLAccess WSDL Interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdair"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                        xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                  xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

<!-- WSDL IMPORTS ################################################## -->
    <wsdl:import location="./wsdai_core_porttypes_0.7.wsdl"
                 namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"/>

<!-- WSDL TYPES #################################################### -->
      <wsdl:types>
        <xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                        elementFormDefault="qualified">
         <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                   schemaLocation="./wsdai_core_types_0.7.xsd" />
         <xsd:include schemaLocation="./wsdair_sql_types_0.7.xsd" />


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

            <!-- general request types -->
            <xsd:complexType name="SQLParameterType">
              <xsd:sequence>
                <xsd:element name="Name" type="xsd:string" />
                <xsd:element name="Value" type="xsd:string"/>
                <xsd:element name="Type" type="xsd:string"/>
              </xsd:sequence>
            </xsd:complexType>

            <xsd:complexType name="SQLExpressionType">
            <xsd:complexContent>
               <xsd:extension base="wsdai:ExpressionType">
                 <xsd:sequence>
                       <xsd:element name="Expression" type="xsd:string"  minOccurs="1" maxOccurs="1"/>
                       <xsd:element name="SQLParameter" type="wsdair:SQLParameterType"  minOccurs="0"
maxOccurs="unbounded"/>
                   </xsd:sequence>
                 </xsd:extension>
            </xsd:complexContent>
            </xsd:complexType>
            <xsd:element name="SQLExpression" type="wsdair:SQLExpressionType" abstract="true" />

      <!-- ################################ -->
      <!-- ### sqlExecute Message Types ### -->
      <!-- ################################ -->

            <xsd:element name="SQLExecuteRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
                          <xsd:element ref="wsdair:SQLExpression"  minOccurs="1" maxOccurs="1"/>
                          <xsd:element ref="wsdai:DatasetType"  minOccurs="0" maxOccurs="1"/>
                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
            </xsd:complexType>
            </xsd:element>

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

      <!-- ####################################### -->
      <!-- ### sqlExecuteFactory Message Types ### -->
      <!-- ####################################### -->
            <xsd:element name="SQLExecuteFactoryRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
```

```
                        <xsd:element ref="wsdair:SQLExpression"  minOccurs="1" maxOccurs="1"/>

                          <xsd:element ref="wsdai:Configuration" minOccurs="0" maxOccurs="1" />
                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>

            <!-- assumes that these messages result in a service/resource that contains all of -->
            <!-- the possible responses from a SQL execute (rowset, count, value, parameter etc) -->
        <xsd:element name="SQLExecuteFactoryResponse" type="wsdai:AddressType" />
      </xsd:schema>
  </wsdl:types>

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

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

    <wsdl:message name="GetSQLPropertyDocumentResponse">
      <wsdl:part name="GetSQLPropertyDocumentResponse"
                 element="wsdair:SQLPropertyDocument" />
    </wsdl:message>

      <!-- ########################### -->
      <!-- ### sqlExecute Messages ### -->
      <!-- ########################### -->
      <wsdl:message name="SQLExecuteRequest">
        <wsdl:part name="SQLExecuteRequest" element="wsdair:SQLExecuteRequest"/>
      </wsdl:message>

      <wsdl:message name="SQLExecuteResponse">
            <wsdl:part name="SQLExecuteResponse" element="wsdair:SQLExecuteResponse"/>
      </wsdl:message>

      <!-- ################################## -->
      <!-- ### sqlExecuteFactory Messages ### -->
      <!-- ################################## -->
      <wsdl:message name="SQLExecuteFactoryRequest">
            <wsdl:part name="SQLExecuteFactoryRequest" element="wsdair:SQLExecuteFactoryRequest"/>
      </wsdl:message>

      <wsdl:message name="SQLExecuteFactoryResponse">
            <wsdl:part name="SQLExecuteFactoryResponse" element="wsdair:SQLExecuteFactoryResponse"/>
      </wsdl:message>

<!-- WSDL PORT TYPES ############################################### -->
      <wsdl:portType name="SQLAccessPT">

        <wsdl:operation name="GetSQLPropertyDocument">
          <wsdl:input  name="GetSQLPropertyDocumentRequest"
                       message="wsdair:GetSQLPropertyDocumentRequest" />
          <wsdl:output name="GetSQLPropertyDocumentResponse"
                       message="wsdair:GetSQLPropertyDocumentResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
        </wsdl:operation>

            <wsdl:operation name="SQLExecute">
                  <wsdl:input message="wsdair:SQLExecuteRequest" />
                  <wsdl:output message="wsdair:SQLExecuteResponse" />
            <wsdl:fault  name="InvalidResourceNameFault"
                         message="wsdai:InvalidResourceNameFault" />
            <wsdl:fault  name="InvalidRequestDocumentFault"
                         message="wsdai:InvalidRequestDocumentFault" />
            </wsdl:operation>

      </wsdl:portType>

      <wsdl:portType name="SQLFactoryPT">

            <wsdl:operation name="SQLExecuteFactory">
                  <wsdl:input message="wsdair:SQLExecuteFactoryRequest" />
                  <wsdl:output message="wsdair:SQLExecuteFactoryResponse" />
            <wsdl:fault  name="InvalidResourceNameFault"
                         message="wsdai:InvalidResourceNameFault" />
            <wsdl:fault  name="InvalidRequestDocumentFault"
                         message="wsdai:InvalidRequestDocumentFault" />
            </wsdl:operation>

      </wsdl:portType>
</wsdl:definitions>
```

Appendix A.2 – SQLAccess XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:wrs="http://java.sun.com/xml/ns/jdbc"
        xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
        xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

      <xsd:import namespace="http://java.sun.com/xml/ns/jdbc"
                schemaLocation="./webrowset-jdbc150.xsd" />

      <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                schemaLocation="./wsdai_core_types_0.7.xsd" />

<!-- properties -->
    <xsd:element name="CIMDescription">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any namespace="http://www.dmtf.org/cim/?????" processContents="strict" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>

<!-- sql access -->
      <xsd:simpleType name="LanguageCapabilitiesType">
            <xsd:union>
                  <xsd:simpleType>
                      <xsd:restriction base="xsd:token">
                          <xsd:enumeration value="SQL92Expression"/>
                          <xsd:enumeration value="SQL99Expression"/>
                          <xsd:enumeration value="SQL03Expression"/>
                      </xsd:restriction>
                  </xsd:simpleType>
                  <xsd:simpleType>
                        <xsd:restriction base="xsd:token"/>
                        <!-- allows for other expression types -->
                  </xsd:simpleType>
          </xsd:union>
      </xsd:simpleType>

    <xsd:element name="LanguageCapabilities" type="wsdair:LanguageCapabilitiesType" />

      <!-- general response types -->
      <xsd:element name="SQLUpdateCount" type="xsd:int" />
      <xsd:element name="SQLOutputParameter" type="xsd:string" />
      <xsd:element name="SQLReturnValue" type="xsd:string" />

      <xsd:complexType name="SQLCommunicationsAreaType">
            <xsd:sequence>
                  <xsd:element name="SQLState" type="xsd:string" />
                  <xsd:element name="VendorCode" type="xsd:string" />
                  <xsd:element name="MessageText" type="xsd:string" />
            </xsd:sequence>
      </xsd:complexType>

      <xsd:element name="SQLCommunicationsArea" type="wsdair:SQLCommunicationsAreaType"/>

      <xsd:complexType name="SQLDatasetType">
           <xsd:complexContent  mixed="true">
              <xsd:extension base="wsdai:DatasetType">
                <xsd:sequence>
                  <xsd:choice maxOccurs="unbounded">
                        <xsd:element ref="wrs:webRowSet" minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element ref="wsdair:SQLUpdateCount" minOccurs="0" maxOccurs="unbounded"/>
                  </xsd:choice>
                  <xsd:element ref="wsdair:SQLOutputParameter" minOccurs="0" maxOccurs="unbounded"/>
                  <!--xsd:element ref="wsdair:SQLReturnValue" minOccurs="0" maxOccurs="unbounded"/-->
                  <xsd:element ref="wsdair:SQLCommunicationsArea" minOccurs="1" maxOccurs="unbounded"/>
                </xsd:sequence>
              </xsd:extension>
          </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="SQLDataset" type="wsdair:SQLDatasetType" substitutionGroup="wsdai:Dataset"/>

<!-- property and configuration documents -->
    <xsd:complexType name="SQLPropertyDocumentType">
```

```
      <xsd:complexContent>
        <xsd:extension base="wsdai:PropertyDocumentType">
            <xsd:sequence>
            <xsd:element ref="wsdair:CIMDescription" />
            <xsd:element ref="wsdair:LanguageCapabilities" />
            </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="SQLPropertyDocument" type="wsdair:SQLPropertyDocumentType"/>

    <xsd:complexType name="SQLPTConfigurationType">
      <xsd:complexContent>
        <xsd:extension base="wsdai:ConfigurationType">
          <xsd:sequence>
            <xsd:element name="PortType">
              <xsd:simpleType>
                <xsd:restriction base="xsd:QName">
                  <xsd:enumeration value="wsdai:SQLAccessPT"/>
                  <xsd:enumeration value="wsdai:SQLFactoryPT"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>
            <xsd:element ref="wsdai:ConfigurationDocument"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="SQLPTConfiguration" type="wsdair:SQLPTConfigurationType"
substitutionGroup="wsdai:Configuration"/>

</xsd:schema>
```

Appendix A.3 – SQLAccess WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  name="daissql"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
```

```
                xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

  <wsdl:import location="./wsdair_sql_porttypes_0.7.wsdl"
               namespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"/>

<wsdl:binding name="SQLAccessSOAP" type="wsdair:SQLAccessPT">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetSQLPropertyDocument">
    <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLPropertyDocument"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SQLExecute">
    <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/SQLExecute"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

<wsdl:binding name="SQLFactorySOAP" type="wsdair:SQLFactoryPT">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="SQLExecuteFactory">
    <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/SQLExecuteFactory"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

<wsdl:service name="SQLService">
```

```
    <wsdl:port binding="wsdair:SQLAccessSOAP" name="SQLAccessSOAP">
      <soap:address location="http://tempuri.org"/>
    </wsdl:port>
    <wsdl:port binding="wsdair:SQLFactorySOAP" name="SQLFactorySOAP">
      <soap:address location="http://tempuri.org"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Appendix B.1 – SQLResponseAccess WSDL Interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdair"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                          xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                          xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

<!-- WSDL IMPORTS ################################################## -->
    <wsdl:import location="./wsdai_core_porttypes_0.7.wsdl"
                 namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"/>

<!-- WSDL TYPES #################################################### -->
    <wsdl:types>

        <xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                      elementFormDefault="qualified">
          <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                    schemaLocation="./wsdai_core_types_0.7.xsd" />
            <xsd:include schemaLocation="./wsdair_sql_types_0.7.xsd"/>
            <xsd:include schemaLocation="./wsdair_response_types_0.7.xsd" />


      <!-- ######################################## -->
      <!-- ### GetSQLResponseItem Message Types ### -->
      <!-- ######################################## -->
            <xsd:element name="GetSQLResponseItemRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
          <xsd:sequence>
                  <xsd:element name="StartPosition" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                    <xsd:element name="Count" type="xsd:int"  minOccurs="0" maxOccurs="1"/>
                  </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="GetSQLResponseItemResponse">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element ref="wsdair:SQLDataset"  minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>

<!-- ################################## -->
<!-- ### GetSQLRowset Message Types ### -->
<!-- ################################## -->
      <xsd:element name="GetSQLRowsetRequest">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:extension base="wsdai:RequestType">
          <xsd:sequence>
                    <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                    <xsd:element ref="wsdai:DatasetType"  minOccurs="0" maxOccurs="1"/>

                  </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>

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

```
<!-- ######################################### -->
<!-- ### GetSQLRowsetFactory Message Types ### -->
<!-- ######################################### -->
      <xsd:element name="GetSQLRowsetFactoryRequest">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:extension base="wsdai:RequestType">
          <xsd:sequence>
                    <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>

                    <xsd:element ref="wsdai:Configuration" minOccurs="0" maxOccurs="1" />
                  </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>

  <xsd:element name="GetSQLRowsetFactoryResponse" type="wsdai:AddressType" />

<!-- ####################################### -->
<!-- ### GetSQLUpdateCount Message Types ### -->
<!-- ####################################### -->
      <xsd:element name="GetSQLUpdateCountRequest">
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:extension base="wsdai:RequestType">
          <xsd:sequence>
                    <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="GetSQLUpdateCountResponse">
            <xsd:complexType>
                  <xsd:sequence>
                        <xsd:element name="UpdateCount" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                  </xsd:sequence>
            </xsd:complexType>
      </xsd:element>
      <!-- ####################################### -->
      <!-- ### GetSQLReturnValue Message Types ### -->
      <!-- ####################################### -->
            <xsd:element name="GetSQLReturnValueRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
                        <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>

            <xsd:element name="GetSQLReturnValueResponse">
                  <xsd:complexType>
                        <xsd:sequence>
                              <xsd:element name="ReturnValue" type="xsd:string"  minOccurs="1"
maxOccurs="1"/>
                        </xsd:sequence>
                  </xsd:complexType>
            </xsd:element>

      <!-- ########################################### -->
      <!-- ### GetSQLOutputParameter Message Types ### -->
      <!-- ########################################### -->
            <xsd:element name="GetSQLOutputParameterRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
                          <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>

            <xsd:element name="GetSQLOutputParameterResponse">
                  <xsd:complexType>
                        <xsd:sequence>
```

```
                              <xsd:element name="OutputParameter" type="xsd:string"  minOccurs="1"
maxOccurs="1"/>
                        </xsd:sequence>
                  </xsd:complexType>
            </xsd:element>

      <!-- ############################################# -->
      <!-- ### GetSQLCommunicationArea Message Types ### -->
      <!-- ############################################# -->
            <xsd:element name="GetSQLCommunicationAreaRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
                          <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>

            <xsd:element name="GetSQLCommunicationAreaResponse">
                  <xsd:complexType>
                        <xsd:sequence>
                              <xsd:element ref="wsdair:SQLCommunicationsArea" minOccurs="1" maxOccurs="1"/>
                        </xsd:sequence>
                  </xsd:complexType>
            </xsd:element>

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

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

      <!-- ####################################### -->
      <!-- ### GetSQLResponsePropertyDocument Messages ### -->
      <!-- ####################################### -->
    <wsdl:message name="GetSQLResponsePropertyDocumentRequest">
      <wsdl:part name="GetSQLResponsePropertyDocumentRequest"
                 element="wsdai:GetDataResourcePropertyDocumentRequest" />
    </wsdl:message>
    <wsdl:message name="GetSQLResponsePropertyDocumentResponse">
      <wsdl:part name="GetSQLResponsePropertyDocumentResponse"
                 element="wsdair:SQLResponsePropertyDocument" />
    </wsdl:message>

      <!-- ################################### -->
      <!-- ### GetSQLResponseItem Messages ### -->
      <!-- ################################### -->
      <wsdl:message name="GetSQLResponseItemRequest">
            <wsdl:part name="GetSQLResponseItemRequest" element="wsdair:GetSQLResponseItemRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLResponseItemResponse">
            <wsdl:part name="GetSQLResponseItemResponse" element="wsdair:GetSQLResponseItemResponse" />
      </wsdl:message>

      <!-- ############################# -->
      <!-- ### GetSQLRowset Messages ### -->
      <!-- ############################# -->
      <wsdl:message name="GetSQLRowsetRequest">
            <wsdl:part name="GetSQLRowsetRequest" element="wsdair:GetSQLRowsetRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLRowsetResponse">
            <wsdl:part name="GetSQLRowsetResponse" element="wsdair:GetSQLRowsetResponse" />
      </wsdl:message>

      <!-- #################################### -->
      <!-- ### GetSQLRowsetFactory Messages ### -->
      <!-- #################################### -->
      <wsdl:message name="GetSQLRowsetFactoryRequest">
            <wsdl:part name="GetSQLRowsetFactoryRequest" element="wsdair:GetSQLRowsetFactoryRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLRowsetFactoryResponse">
            <wsdl:part name="GetSQLRowsetFactoryResponse" element="wsdair:GetSQLRowsetFactoryResponse" />
      </wsdl:message>

      <!-- #################################### -->
      <!-- ### GetSQLUpdateCount Messages   ### -->
      <!-- #################################### -->
      <wsdl:message name="GetSQLUpdateCountRequest">
```

```
            <wsdl:part name="GetSQLUpdateCountRequest" element="wsdair:GetSQLUpdateCountRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLUpdateCountResponse">
            <wsdl:part name="GetSQLUpdateCountResponse" element="wsdair:GetSQLUpdateCountResponse" />
      </wsdl:message>

    <!-- #################################### -->
      <!-- ### GetSQLReturnValue Messages   ### -->
      <!-- #################################### -->
      <wsdl:message name="GetSQLReturnValueRequest">
            <wsdl:part name="GetSQLReturnValueRequest" element="wsdair:GetSQLReturnValueRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLReturnValueResponse">
            <wsdl:part name="GetSQLReturnValueResponse" element="wsdair:GetSQLReturnValueResponse" />
      </wsdl:message>

    <!-- ######################################## -->
      <!-- ### GetSQLOutputParameter Messages   ### -->
      <!-- ######################################## -->
      <wsdl:message name="GetSQLOutputParameterRequest">
            <wsdl:part name="GetSQLOutputParameterRequest" element="wsdair:GetSQLOutputParameterRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLOutputParameterResponse">
            <wsdl:part name="GetSQLOutputParameterResponse" element="wsdair:GetSQLOutputParameterResponse"
/>
      </wsdl:message>

    <!-- ########################################## -->
      <!-- ### GetSQLCommunicationArea Messages   ### -->
      <!-- ########################################## -->
      <wsdl:message name="GetSQLCommunicationAreaRequest">
            <wsdl:part name="GetSQLCommunicationAreaRequest"
element="wsdair:GetSQLCommunicationAreaRequest" />
      </wsdl:message>

      <wsdl:message name="GetSQLCommunicationAreaResponse">
            <wsdl:part name="GetSQLCommunicationAreaResponse"
element="wsdair:GetSQLCommunicationAreaResponse" />
      </wsdl:message>
```

```
<!-- WSDL PORT TYPES ############################################### -->
      <wsdl:portType name="ResponseAccessPT">

        <wsdl:operation name="GetSQLResponsePropertyDocument">
          <wsdl:input  name="GetSQLResponsePropertyDocumentRequest"
                       message="wsdair:GetSQLResponsePropertyDocumentRequest" />
          <wsdl:output name="GetSQLResponsePropertyDocumentResponse"
                       message="wsdair:GetSQLResponsePropertyDocumentResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
        </wsdl:operation>

            <wsdl:operation name="GetSQLResponseItem">
              <wsdl:input message="wsdair:GetSQLResponseItemRequest" />
              <wsdl:output message="wsdair:GetSQLResponseItemResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

            <wsdl:operation name="GetSQLRowset">
              <wsdl:input message="wsdair:GetSQLRowsetRequest" />
              <wsdl:output message="wsdair:GetSQLRowsetResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

            <wsdl:operation name="GetSQLUpdateCount">
              <wsdl:input message="wsdair:GetSQLUpdateCountRequest" />
              <wsdl:output message="wsdair:GetSQLUpdateCountResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

            <wsdl:operation name="GetSQLReturnValue">
              <wsdl:input message="wsdair:GetSQLReturnValueRequest" />
              <wsdl:output message="wsdair:GetSQLReturnValueResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

            <wsdl:operation name="GetSQLOutputParameter">
```

```
              <wsdl:input message="wsdair:GetSQLOutputParameterRequest" />
              <wsdl:output message="wsdair:GetSQLOutputParameterResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

            <wsdl:operation name="GetSQLCommunicationArea">
              <wsdl:input message="wsdair:GetSQLCommunicationAreaRequest" />
              <wsdl:output message="wsdair:GetSQLCommunicationAreaResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

      </wsdl:portType>

      <wsdl:portType name="ResponseFactoryPT">

            <wsdl:operation name="GetSQLRowsetFactory">
              <wsdl:input message="wsdair:GetSQLRowsetFactoryRequest" />
              <wsdl:output message="wsdair:GetSQLRowsetFactoryResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

      </wsdl:portType>

</wsdl:definitions>
```

Appendix B.2 – SQLResponseAccess XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:wrs="http://java.sun.com/xml/ns/jdbc"
        xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
        xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

  <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
              schemaLocation="./wsdai_core_types_0.7.xsd" />

<!-- sql response description -->
  <xsd:complexType name="SQLResponseItemType">
```

```
    <xsd:sequence>
      <xsd:element name="SQLResponseItemSequenceNumber" type="xsd:int" />
      <xsd:element name="SQLResponseItemFormatType" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="SQLResponseItem" type="wsdair:SQLResponseItemType"/>
  <xsd:element name="NumberOfSQLRowsets" type="xsd:int" />
  <xsd:element name="NumberOfSQLUpdateCounts" type="xsd:int" />
  <xsd:element name="NumberOfSQLReturnValues" type="xsd:int" />
  <xsd:element name="NumberOfSQLOutputParameters" type="xsd:int" />
  <xsd:element name="NumberOfSQLCommunicationsAreas" type="xsd:int" />

<!-- properties that describe a SQL response data service -->
  <xsd:complexType name="SQLResponsePropertyDocumentType">
      <xsd:complexContent>
        <xsd:extension base="wsdai:PropertyDocumentType">
            <xsd:sequence>
                <xsd:element ref="wsdair:SQLResponseItem" minOccurs="1" maxOccurs="1"/>

                  <xsd:element ref="wsdair:NumberOfSQLRowsets" minOccurs="1" maxOccurs="1"/>
                  <xsd:element ref="wsdair:NumberOfSQLUpdateCounts" minOccurs="1" maxOccurs="1"/>

                  <xsd:element ref="wsdair:NumberOfSQLReturnValues" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="wsdair:NumberOfSQLOutputParameters" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="wsdair:NumberOfSQLCommunicationsAreas" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="SQLResponsePropertyDocument" type="wsdair:SQLResponsePropertyDocumentType"/>

   <xsd:complexType name="ResponsePTConfigurationType">
      <xsd:complexContent>
        <xsd:extension base="wsdai:ConfigurationType">
          <xsd:sequence>
            <xsd:element name="PortType">
              <xsd:simpleType>
                <xsd:restriction base="xsd:QName">
                  <xsd:enumeration value="wsdai:ResponseAccessPT"/>
                  <xsd:enumeration value="wsdai:ResponseFactoryPT"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>
            <xsd:element ref="wsdai:ConfigurationDocument"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="ResponsePTConfiguration" type="wsdair:ResponsePTConfigurationType"
substitutionGroup="wsdai:Configuration"/>
</xsd:schema>
```

Appendix B.3 – SQLResponseAccess WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  name="daisresponse"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                  xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                  xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

    <wsdl:import location="./wsdair_response_porttypes_0.7.wsdl"
                 namespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"/>

  <wsdl:binding name="ResponseAccessSOAP" type="wsdair:ResponseAccessPT">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetSQLResponsePropertyDocument">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-
DAIR/GetSQLResponsePropertyDocument"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLResponseItem">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLResponseItem"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLRowset">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowset"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLUpdateCount">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowsetFactory"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLReturnValue">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowsetFactory"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLOutputParameter">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowsetFactory"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSQLCommunicationArea">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowsetFactory"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:binding name="ResponseFactorySOAP" type="wsdair:ResponseFactoryPT">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetSQLRowsetFactory">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetSQLRowsetFactory"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="ResponseService">
    <wsdl:port binding="wsdair:ResponseAccessSOAP" name="ResponseAccessSOAP">
      <soap:address location="http://tempuri.org"/>
    </wsdl:port>
    <wsdl:port binding="wsdair:ResponseFactorySOAP" name="ResponseFactorySOAP">
      <soap:address location="http://tempuri.org"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Appendix C.1 – SQLRowSetAccess WSDL Interfaces

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="wsdair"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                          xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                          xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">
```

```
<!-- WSDL IMPORTS ################################################## -->
    <wsdl:import location="./wsdai_core_porttypes_0.7.wsdl"
                 namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"/>

<!-- WSDL TYPES #################################################### -->
    <wsdl:types>
        <xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                      elementFormDefault="qualified">
        <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                    schemaLocation="./wsdai_core_types_0.7.xsd" />
        <xsd:include schemaLocation="./wsdair_sql_types_0.7.xsd" />
            <xsd:include schemaLocation="./wsdair_rowset_types_0.7.xsd" />


      <!-- ############################### -->
      <!-- ### GetTuples Message Types ### -->
      <!-- ############################### -->
            <xsd:element name="GetTuplesRequest">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:extension base="wsdai:RequestType">
                <xsd:sequence>
                          <xsd:element name="StartPosition" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                          <xsd:element name="Count" type="xsd:int"  minOccurs="1" maxOccurs="1"/>
                          <xsd:element ref="wsdai:DatasetType"  minOccurs="0" maxOccurs="1"/>

                        </xsd:sequence>
              </xsd:extension>
            </xsd:complexContent>
              </xsd:complexType>
            </xsd:element>

            <xsd:element name="GetTuplesResponse">
                  <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 ################################################# -->

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

    <wsdl:message name="GetRowsetPropertyDocumentResponse">
      <wsdl:part name="GetRowsetPropertyDocumentResponse"
                 element="wsdair:RowsetPropertyDocument" />
    </wsdl:message>

      <!-- ######################### -->
      <!-- ### GetTuples Messages ### -->
      <!-- ######################### -->
      <wsdl:message name="GetTuplesRequest">
            <wsdl:part name="GetTuplesRequest" element="wsdair:GetTuplesRequest" />
      </wsdl:message>

      <wsdl:message name="GetTuplesResponse">
            <wsdl:part name="GetTuplesResponse" element="wsdair:GetTuplesResponse" />
      </wsdl:message>

<!-- WSDL PORT TYPES ############################################### -->
      <wsdl:portType name="RowsetAccessPT">

        <wsdl:operation name="GetRowsetPropertyDocument">
          <wsdl:input  name="GetRowsetPropertyDocumentRequest"
                       message="wsdair:GetRowsetPropertyDocumentRequest" />
          <wsdl:output name="GetRowsetPropertyDocumentResponse"
                       message="wsdair:GetRowsetPropertyDocumentResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
        </wsdl:operation>

            <wsdl:operation name="GetTuples">
            <wsdl:input message="wsdair:GetTuplesRequest" />
              <wsdl:output message="wsdair:GetTuplesResponse" />
          <wsdl:fault  name="InvalidResourceNameFault"
                       message="wsdai:InvalidResourceNameFault" />
            </wsdl:operation>

      </wsdl:portType>

</wsdl:definitions>
```

Appendix C.2 – SQLRowSetAccess XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:wrs="http://java.sun.com/xml/ns/jdbc"
        xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
        xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

  <xsd:import namespace="http://java.sun.com/xml/ns/jdbc"
              schemaLocation="./webrowset-jdbc150.xsd" />
  <xsd:import namespace="http://www.ggf.org/namespaces/2005/06/WS-DAI"
              schemaLocation="./wsdai_core_types_0.7.xsd" />

<!-- rowset description -->
  <xsd:complexType name="RowSchemaType">
    <xsd:sequence>
      <xsd:element ref="wrs:metadata"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="RowSchema" type="wsdair:RowSchemaType" />

  <xsd:element name="NoOfRows" type="xsd:int" />

<!-- rowset access -->
  <xsd:element name="AccessMode">
    <xsd:simpleType>
      <xsd:restriction base="xsd:token">
          <xsd:enumeration value="Forward"/>
          <xsd:enumeration value="Random"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="CursorHeldOverTxnBoundary" type="xsd:boolean" />

  <!-- the properties that describe a rowsetservice -->
  <xsd:complexType name="RowsetPropertyDocumentType">
    <xsd:complexContent>
      <xsd:extension base="wsdai:PropertyDocumentType">
          <xsd:sequence>
            <xsd:element ref="wsdair:RowSchema" minOccurs="1" maxOccurs="1"/>
              <xsd:element ref="wsdair:NoOfRows" minOccurs="1" maxOccurs="1" />
          <xsd:element ref="wsdair:AccessMode" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="wsdair:CursorHeldOverTxnBoundary" minOccurs="1" maxOccurs="1"/>
          </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="RowsetPropertyDocument" type="wsdair:RowsetPropertyDocumentType"/>

    <xsd:complexType name="RowsetPTConfigurationType">
      <xsd:complexContent>
        <xsd:extension base="wsdai:ConfigurationType">
          <xsd:sequence>
            <xsd:element name="PortType">
              <xsd:simpleType>
                <xsd:restriction base="xsd:QName">
                  <xsd:enumeration value="wsdai:RowsetAccessPT"/>
                  <xsd:enumeration value="wsdai:RowsetFactoryPT"/>
                </xsd:restriction>
              </xsd:simpleType>
            </xsd:element>
            <xsd:element ref="wsdai:ConfigurationDocument"/>
            <xsd:element ref="wsdair:AccessMode" minOccurs="0" maxOccurs="1"/>
              <xsd:element ref="wsdair:CursorHeldOverTxnBoundary" minOccurs="0" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="RowsetPTConfiguration" type="wsdair:RowsetPTConfigurationType"
substitutionGroup="wsdai:Configuration"/>

</xsd:schema>
```

Appendix C.3 – SQLRowSetAccess WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  name="daisrowset"
                  targetNamespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"
                  xmlns:wsdai="http://www.ggf.org/namespaces/2005/06/WS-DAI"
                  xmlns:wsdair="http://www.ggf.org/namespaces/2005/06/WS-DAIR">

    <wsdl:import location="./wsdair_rowset_porttypes_0.7.wsdl"
                 namespace="http://www.ggf.org/namespaces/2005/06/WS-DAIR"/>

  <wsdl:binding name="RowsetAccessSOAP" type="wsdair:RowsetAccessPT">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetRowsetPropertyDocument">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-
DAIR/GetRowsetPropertyDocument"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetTuples">
      <soap:operation soapAction="http://www.ggf.org/namespaces/2005/06/WS-DAIR/GetTuples"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>

  </wsdl:binding>
  <wsdl:service name="RowsetService">
    <wsdl:port binding="wsdair:RowsetAccessSOAP" name="RowsetAccessSOAP">
      <soap:address location="http://tempuri.org"/>
    </wsdl:port>
  </wsdl:service>
```

```
</wsdl:definitions>
```