

Design and Implementation of OGSA-DAI-RDF

Isao Kojima
kojima@ni.aist.go.jp

Data Grid Team
Grid Technology Research Center
AIST, Japan

Abstract

This paper presents the OGSA-DAI-RDF middleware that extends OGSA-DAI access to RDF database systems, e.g., Sesame and Jena. Several OGSA-DAI activities for handling RDF data and ontology are implemented. The query language interface is based on SPARQL query language.

Introduction

The National Institute of Advanced Science and Technology (AIST) of Japan started a 5-year project called AIST-SOA (Service Oriented Architecture). One aim of the project is to develop service-based middleware that utilizes Grid technology and Semantic Web technology. To support the AIST-SOA data platform, RDF (Resource Description Framework), which is a common metadata representation format used in semantic web applications, is very important. For example, we have conducted the following RDF-related grid research as a part of AIST-SOA.

1. **Distributed RDF processing that provides scalable RDF storage:** One approach^[8] is to use P2P technology like a DHT (distributed hash table) and support RDF query processing like SPARQL^[5] and RDQL.
2. **High-level resource management that uses Semantic Web technology:** For example, Semantic MDS^[7] supports the ontology for grid resource management and discovery. This system uses OWL-S and RDF storage as the resource management database.

Therefore, it is very important to provide a service-based interface for RDF databases.

To develop the database middleware, we already have several OGSA-DAI^[1] based projects, such as OGSA-WebDB^[9]; it is a very natural process to extend the OGSA-DAI interface to support RDF databases.

2. OGSA-DAI-RDF

2.1 RDF-handling Activities

OGSA-DAI provides a framework called an Activity for extending the system. There are several kinds of Activities for relational databases, XML databases, data transport, and data conversion. Using this framework, we implemented a family of RDF-handling Activities, including the following examples:

- 1) SPARQL QueryStatement Activity
- 2) RDF Bulkload Activity
- 3) RDF ResourceManagement Activity
- 4) RDF CollectionManagement Activity

For instance, the SPARQL QueryStatement Activity receives the W3C SPARQL representation and gets the result in RDF XML format. W3C also discusses the SPARQL protocol^[6], and the message format used for OGSA-DAI-RDF is designed to be similar to those specifications. An example PerformDocument for SPARQL is shown in Fig.1.

ResourceManagement Activity provides create and delete functions to manipulate RDF statements. CollectionManagement Activity supports create, delete and list functions to RDF data repository(Model for Jena).

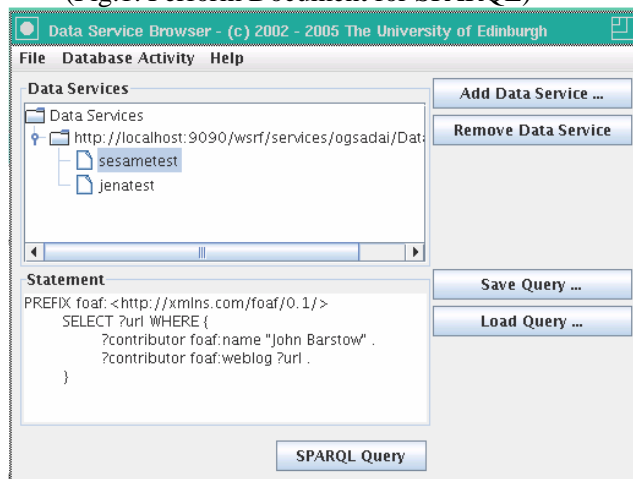
Our current implementation supports Jena^[3] and Sesame^[4]. Activities are designed to have as much common interface as possible between these products. Fig.2 shows the data service instance which has Sesame and Jena resources.

```

<perform xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
  <documentation> This is a SPARQL Performdoc example.</documentation>
</deliverFromFile name="myDelivery">
  <fromFile>/home/globus/ce-wsrf-schema.owl</fromFile>
  <toLocal name="owlSchema" />
</deliverFromFile>
<rdfCollectionManagement name="myActivity1">
  <createCollection name="memory" />
  <outputStream name="memoryObject" />
</rdfCollectionManagement>
<rdfResourceManagement name="myActivity2">
  <createResource fromData="owlSchema" toStream="memoryObject" />
  <outputStream name="memoryObjectSchema" />
</rdfResourceManagement>
<sparqlQueryStatement name="myActivity3">
  <queryRequest stream="memoryObjectSchema">
    PREFIX semds:<http://www.aist.go.jp/schema/ce-wsrf-schema.owl#>
    PREFIX glue:<http://mds.globus.org/glue/ce/1.1#>
    PREFIX cpu:<http://www.aist.go.jp/namespaces/cpu#>
    PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX org:<http://www.aist.go.jp/namespaces/org#>
    SELECT ?hostName ?procLoadVal ?ramVal WHERE{ ?host
      glue:hasProcessorLoad ?procLoad . ?procLoad glue:hasProcLoad-1 ?load1 . ?load1
      semds:value ?procLoadVal . ?host glue:hasHostName ?hostName . ?host
      glue:hasMainMemory ?mainMemory . ?mainMemory
      glue:hasRAMAvailable ?ramAvail . ?ramAvail semds:value ?ramVal . ?host
      glue:hasProcessor ?proc . ?proc semds:hasSemantics ?procType . ?procType rdf:type
      cpu:Sun . ?host semds:hasSemantics ?se . ?se rdf:type org:GTRC }</queryRequest>
  </sparqlQueryStatement>
  <queryResponse name="JenaTestOutput" />
</deliverToFile name="deliverResults">
  <fromLocal from="JenaTestOutput" />
  <toFile>/tmp/results.txt</toFile>
</deliverToFile>
</perform>

```

(Fig.1. Perform Document for SPARQL)



(Fig. 2. Extended Databrowser to support SPARQL)

Using these activities, it is possible to manipulate

individual RDF statements/graphs.

2.2 Ontology Handling Activity

We also implemented several activities to handle ontology data. These activities are based on the reasoning function of Jena and supports OWL and RDFS. Ontology can be specified as an option of SPARQL QueryStatement Activity. It supports activity so that it is possible to pass ontology data from other activities.

2.2 Configuration File Support

In order to extend the OGSA-DAI, we developed a new data resource type called RDF. The extended configuration file of OGSA-DAI (data.service.resource.properties) supports the following examples.

```

## 1-Select the name of the data service resource.
## The default is "DataServiceResource."
## You can change this if you want.

dai.resource.id=sesametest

## 2-Select the type of data resource that forms the data service
## resource.
## Remove the hash (#) from the front of the desired type.
## Only remove one hash!

# dai.data.resource.type=Relational
#dai.data.resource.type=XML
# dai.data.resource.type=Files

dai.data.resource.type=RDF

```

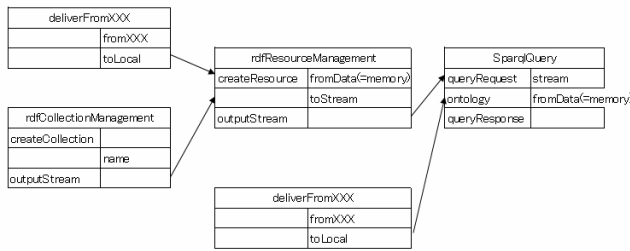
(Fig. 3. Part of the configuration File for RDF databases)

In this configuration, users can specify the RDF storage product. For Jena, it is possible to make RDF repository both on the RDB and on main memory. For Sesame, Sesame has its own tomcat-based http interface; however, we access Sesame directly, so that there is no other tomcat container.

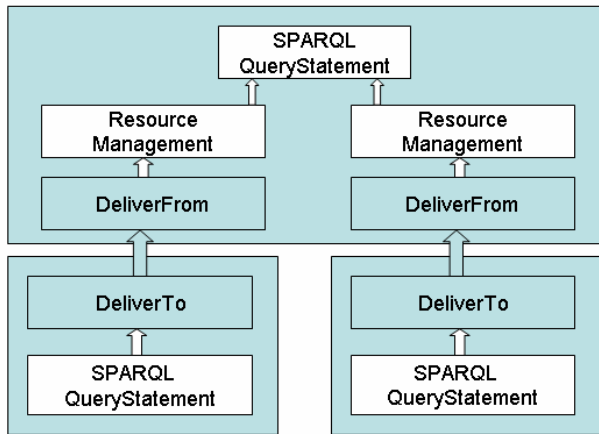
2.3 Distributed RDF Processing based on Activity Programming

Based on the architecture of OGSA-DAI, it is possible to combine other data-processing activities with RDF processing. For example, it is possible to convert the SPARQL result with the XSLTransform activity. Fig.4. shows the workflow of activities.

We are implementing a simple distributed processing service on top of this software. This is also done with the Activity programming of OGSA-DAI framework.



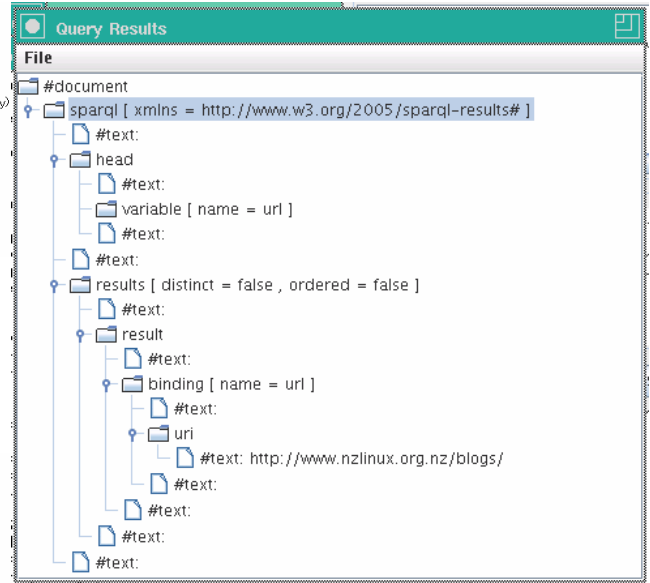
- 1) Create a RDF resource and inserts data from other activity.
 - 2) Query is issued to the resource using the ontology data which is also delivered from other activity
- (Fig.4.1 Activity workflow example)



(Fig.4.2. Simple RDF Distributed Processing Example)

2.4 Extended Data Browser

OGSA-DAI's data browser is extended to support RDF-type data resources. The browser can also supports other resource type and if one selects resources with RDF type, the query interface will change to submit a SPARQL Query. The resulting format of the query is represented in tree/folder format as shown in Fig.5.



(Fig. 5. Folder view of XML of a SPARQL-result format)

3. Status and Future Directions

The first version of OGSA-DAI-RDF has been implemented on top of OGSA-DAI 2.1, and 1Q will be released in 2006. Online demonstration will be presented at the workshop. We are aiming to make a reference implementation for the coming RDF realization of GGF DAIS standard specifications.

4. References

- [1] <http://www.ogsadai.org.uk/>
- [2] <http://www.gtrc.aist.go.jp/dbgrid/ogsawebdb/>
- [3] <http://jena.sourceforge.net/>
- [4] <http://www.openrdf.org/>
- [5] <http://www.w3.org/TR/rdf-sparql-query/>
- [6] <http://www.w3.org/2001/sw/DataAccess/>
- [7] S. Mirza and I. Kojima "Towards Automatic Service Discovery and Monitoring in WS-Resource Framework," 1st SKG Conference, Beijing, 2005.11.
- [8] A. Matono, S. Mirza, and I. Kojima "P2P based distributed query processing based on 3-dimensional cube index," DeWeb2005. Tokyo, 2005.11. (In Japanese)
- [9] I. Kojima and S. Mirza "Design and Implementation of OGSA-WebDB," GGF10 Data Area Workshop, Berlin, 2004.03