Service Data

CSE 487/587
April 20, 2005

References:
Gottschalk's tutorial on Grid Services

Introduction
- Service Data is associated with the instance of a Grid Service.
- Data stored in Service Data Elements can be of two types:
  - Service Metadata: System data, supported interfaces, Cost of using the service, etc.
  - State Information: Operation results, intermediate results, runtime information, etc.

Example
- The example below shows the all-famous MathService with two SDEs:
  - SystemInfo: Service Metadata (System Data)
  - LastResults: State Information (Operation Results)

Another Example
Defining Service Data Elements

- The actual structure of the SDE is defined in an XSD file.
- This file is then imported into the service's gwsdl.
- SDEs can be of simple (LastResult) or complex (SystemInfo) type.
- One or more SDEs can be defined in the PortType definition (in the gwsdl).

---

Defining Service Data Elements

- The definition in the PortType looks as follows:

```xml
<soap:service name="DefaultService">
  <soap:operation name="MethodName">
    <soap:input message="tns:InputMessage"/>
    <soap:output message="tns:OutputMessage"/>
  </soap:operation>
</soap:service>
```

---

Java file (stub)

```java
import java.util.List;

public class DefaultService {
    public voidMethodName(List<InputMessage> inputMessage) {
        // implementation...
    }
}
```

---

Defining Service Data Elements

- The following attributes refer to various properties of the SDE:
  - minOccurs: The minimum number of values that this SDE can have.
  - maxOccurs: The maximum number of values that this SDE can have. The value of this attribute can be unbounded, which indicates an array with no size limit.
  - nillable: True or false. Specifies if the value of this SDE can be changed by a client.
  - modifiable: True or false. Specifies if the value of this SDE can be NULL.
  - extendable: This attribute can have the following values:
    - static: The value of the SDE is provided in the GWSDL description.
    - constant: The value of the SDE is set when the Grid Service is created, but remains constant after that.
    - extendable: New elements can be added to the SDE, but not removed.
    - mutable: New elements can be added and removed.
Defining Service Data Elements

- Besides the above additions to the gwsdl there are a few more (import XSD file and inclusion of additional namespaces). Look at:
  $TUTORIAL_DIR/schema/<username>_progtutorial/
  MemoryMonitorService/MemoryMonitor.gwsdl
- An additional namespace to package mapping must be setup in the namespace2package.mappings file to map the new service data namespace

Service implementation

- Two new objects are required for dealing with Service Data:
  `private ServiceData mathDataSDE;`
  `private MathDataType mathDataValue;`
- The first one refers to the SDE and the second one refers to the value of the SDE
- SDEs are created in the `postCreate` method of the service class. This method gets called after the creation of the service instance.

---

Service Implementation

```java
public void postCreate(ObjectContext context) throws ServiceException {
    // Create Service Data Element
    memoryData = this.getMemoryData.Tab(); createMemoryData();
    // Create a MathDataType (instance and set initial value
    memoryDataValue = new MathDataType();
    memoryDataValue.setData((String)context.getValue("initialValue"));
    // Set the instance = runtime.getOutInst();
    public void exist()
    {
        memoryData.setMemoryData.DiskMemory();
        memoryData.setMemoryData.LinuxMemory();
        memoryData.setMemoryData.cppmemDirector();
    }, 0, 1000)
    // Set the value of the SDE to the MemoryDataType instance
    memoryDataValue.setMemoryData();
    // Add SDE to Service Data
    this.setServiceData(Set<MemoryDataValue>);
}
```

---

Client

```java
// set a reference to the MemoryDataType
MemoryDataMap memoryDataMap = new MemoryDataMap();
MemoryDataType memoryDataType = memoryDataMap.getServiceData();
MemoryDataMemory memoryDataMemory = memoryDataMap.getServiceData();
MemoryDataMemory memoryDataMemory = memoryDataMap.getServiceData();
// Get Service Data (memory.getValue("memoryData");
```

```java
MemoryDataMemory memoryDataMemory = MemoryDataMemory.getInstance().newInstance(memoryDataType, MemoryDataType.class);
// Other service data
```
Client

- We get the SDE called MathData using a GridService method called findServiceData, and a 'helper' class to resolve the name into something the Grid Service can understand.
- The findServiceData doesn't return a MathDataType class, but an ExtensibilityType class. This needs to be cast into a MathDataType using more helper classes.
- This returns a MathDataType object. We can use it like any other local object