A Virtual Data Product Toolkit Based on Geospatial Web Service Orchestration

Peisheng Zhao, Liping Di, Yaxing Wei
Center for Spatial Information Science and System
George Mason University

Geoinformatics 2006
Agenda

• Introduction
• Life cycle of virtual data product
• OGC Catalog Service for Web (CSW)
• Model designer
• Virtual data service
• BPELPower – service chain engine
• Conclusion
Introduction

• Virtual Data Product
  – Much geospatial scientific data is not obtained directly from measurements but rather derived from other data by the application of computational procedures.
    • Landslide susceptibility (slope, aspect, NDVI...)

• Geospatial Model
  – Abstract computational procedure
  – Behavior-based high level geospatial knowledge.
Introduction

• Web Service
  – “A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.”
  (http://www.w3.org/TR/ws-arch/)

• Web Service Orchestration
  – Assembling individual Web Services into a service chain (representing a more complicated geospatial model and process flow) to achieve desired results
Introduction

• Toolkit
  – With geospatial knowledge
    • Design
    • Discovery
    • Retrieve
    • Chain
    • Orchestrate
    • Visualization
Life Cycle of Virtual Data Products

Knowledge:
- Geospatial Model

Information:
- Logical Workflow
- Concrete Workflow

Data:
- Workflow Execution
- User Data

XML code snippet:
```xml
<?xml version="1.0" encoding="utf-8"?>
<logicprocess name="Landslide susceptibility">
  <activityname "Land operations">Landslide Susceptibility</activityname>
  <activityname "Slope operations">Slope Susceptibility</activityname>
  <activityname "Aspen operations">Aspen Susceptibility</activityname>
  <activityname "California_WHR13_0">California_WHR13_0</activityname>
  <activityname "NDVI">NDVI</activityname>
</logicprocess>
```

Geoinformatics 2006
OGC Catalog Service for Web (CSW)

• **Directory Role**
  – Providers advertise the availability of their resources, and consumers can then query the metadata to discover and run-time access them.

• **EB/RIM Information Model**
  – Specifies formally how domain objects are organized, constrained and interpreted based on domain conceptual structure.

• **Standard Interfaces**
  – GetCapabilities, describeType, getRecord…

• [http://geobrain.laits.gmu.edu/csw/discovery/](http://geobrain.laits.gmu.edu/csw/discovery/)
Model Designer

• Data Type
  – Data type is a class of data having the same scientific meaning.
  – The data type hierarchical list is chosen according to the scientific terminology of discipline domain.
  – Each data registered in the catalogue is associated with some data types.
Service Type

- Service type is a class of services having the same input data types and output data type.
- Each service registered in the catalogue is associated with a service type.
Model Designer

- Example
  - Select “Landslide_Susceptibility” data type.
  - Find a service type whose output type is “Landslide_Susceptibility”. This is done automatically by the designer. Only the satisfied service types are listed and selectable.
  - Select “Landslide_Susceptibility_2i” service type whose input data types are “Terrain_Slope” and “Terrain_Aspect”. If there are more than one satisfied services, the designer allows the user to view their metadata to assist selection.
  - Find service types whose output data types are “Terrain_Slope” and “Terrain_Aspect” as step 2. Select “Slope” and “Aspect” service type whose input data type is “Terrain_Elevation”.

Geoinformatics 2006
Model Designer

- http://laits.gmu.edu/vdp/

```xml
<logicprocess name="test" targetnamespace="http://bpel.laits.gmu.edu/bpel/lpm">
  <activitymembers>
    <activitymember name="Landslide_Susceptibility_2i" operation="Landslide_Susceptibility_2i" />
    <activitymember name="Slope" operation="Slope" />
    <activitymember name="Aspect" operation="Aspect" />
  </activitymembers>
  <compositeactivity>
    <parallelactivity name="Activity_1" memberfirst="Slope" memberlast="Aspect" />
    <sequenceactivity name="Activity_2" memberfirst="Activity_1" memberlast="Landslide_Susceptibility_2i" />
    <flowactivity name="model" expression="Activity_2" />
  </compositeactivity>
</logicprocess>
```
Virtual Data Service

• **Instantiation**
  – Input: logical workflow
  – Output: BPEL process
  – Service discovery
    • *Find a service instance for each service type in logical workflow*
  – Data service add-in
    • *Find the most appropriate existing data (at the bottom of geospatial model) and add the relevant data services automatically, such as Web Coverage Service, Web Coordinate Transformation Service.*

• **BPEL4WS (Business Process Execution Language for Web Service)**
  – A language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions.
Virtual Data Service

• Standard interfaces
  – OGC Web Coverage Service
    • getCoverage

BPELPower – Service Chain Engine

• Based on the mainstream standards
  – BPEL, WSDL, WSIF, Xalan, Xerces, UDDI, AXIS, SOAP, JNDI, J2EE (servlets/EJBs/JSPs), Jetspeed (Portlets) and JMX. It runs on top of popular application servers, such as Tomcat, J2EE, JBoss, Weblogic and WebSphere.

• “Deploy it”.
  – WSDL-based web services and BEPLE-based web services chain can be deployed in BPELPower, where their validations are checked.

• “Try it”.
  – WSDL-based web services and BEPLE-based web services chain can be executed in BPELPower dynamically. Different invocations (e.g., HTTP POST/GET, SOAP document/rpc, etc.) are well supported.

# BPELPower – Service Chain Engine

## Deployed WSDL Services

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Service Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrencyExchangeService</td>
<td>EMBL Nucleotide Sequence Web Service</td>
</tr>
<tr>
<td>GMU-NGA-WCS</td>
<td>GMU-WICS</td>
</tr>
<tr>
<td>Grass General Commands Service</td>
<td>Grass Imagery Commands Service</td>
</tr>
<tr>
<td>Grass Raster3D Commands Service</td>
<td>Grass Raster Commands Service</td>
</tr>
<tr>
<td>Grass Raster InOut Commands Service</td>
<td>Grass Vector Commands Service</td>
</tr>
<tr>
<td>Grass Vector InOut Commands Service</td>
<td>GridSlope Aspect Service</td>
</tr>
<tr>
<td>GridSlopeService</td>
<td>GridWCS Service Lait</td>
</tr>
<tr>
<td>GridWCS Service</td>
<td>HGPWCS Service</td>
</tr>
</tbody>
</table>

Logged to domain: **default**
Conclusion

• **Life cycle of virtual geospatial product**
  – Design (knowledge) \(\rightarrow\) instantiation (information)
    \(\rightarrow\) execution (data)

• **Interoperability**
  – Web Services
    • *Catalog service, Virtual data service, BPELPower*...

• **Expansibility**
  – Web Service chain
    • *Easily add new services to represent more complex product.*