A Grid Workflow Infrastructure

Dieter Cybok
Consultant
msg systems

GGF10
Berlin, Germany
Tuesday, March 9th, 2004
Motivation

• Scientific applications often require the creation of complex collaborative workflows
• Many e-Scientists lack the necessary low-level expertise to utilize the current generation of Grid toolkits
• Documented workflow specification is beneficial for modelling and managing scientific processes; processes can be easily reused, modified and shared

➡ Making live of e-Scientists easier
Grid Workflow - Patterns

- **Workflow Patterns**
  - Reusing results from workflow research
  - Defining relevant set of workflow patterns relevant to e-Science applications

**Sequence**, **Parallel Split**, **Loop**, **Multi Choice**
Grid Workflow - Approaches

- Orchestration
  - Describes business logic and execution order
  - Executable processes
  - One central workflow engine

![Diagram showing Grid Workflow with activities and services connected through WSDL.]
Grid Workflow - Approaches

• Choreography
  – Sequence of messages that involve multiple services
  – Public message exchanges that occur between Grid services
  – Services involved describe the part they play in the interaction
Grid Workflow - Requirements

• Web services workflow management requirements:
  – Managing transactional integrity
  – Compensating transactions
  – Managing exceptions

• Additional grid workflow requirements:
  – Dealing with large amounts of data
  – Life cycle management
Grid Workflow Infrastructure

Grid Workflow
Workflow Execution
Language

Workflow Engine

GT3 Technology
Grid Workflow Execution Language

- We have considered both, BPEL4WS and WSCI as the base for GWEL
- Reasons for choosing orchestration/BPEL4WS:
  - Definition of end-to-end processes
  - Existing Grid services can be used
  - Central workflow engine
- GWEL:
  - XML based
  - Elements and concepts of BPEL4WS are reused
# Grid Workflow Execution Language

<table>
<thead>
<tr>
<th>GWEL definition</th>
<th>Name, Target Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory Model</td>
<td>List of Factories: Name, Handle</td>
</tr>
<tr>
<td>Data Model</td>
<td>List of Data Sources: Name, Handle</td>
</tr>
<tr>
<td>Variables</td>
<td>List of Variables: Name, messageType</td>
</tr>
<tr>
<td>Fault Handling Model</td>
<td>List of Fault Handlers: Name, Variable</td>
</tr>
<tr>
<td></td>
<td>Activity Model</td>
</tr>
<tr>
<td>Instance Lifecycle Model</td>
<td>Instance Creation: List of Instance Creators: Name, Factory</td>
</tr>
<tr>
<td></td>
<td>Event Handling: Events: List of Events: Type, PortTypeName, Operation</td>
</tr>
<tr>
<td></td>
<td>Activity Model</td>
</tr>
<tr>
<td>Activity Model</td>
<td>Control Flow: List of Activities: Name, portType, operation, variable, data source, data sink</td>
</tr>
</tbody>
</table>
• Architecture is based on the workflow reference model
• Prototypical implementation with Java
Underlying Technology

• OGSA
  – Concept for a framework of services that support Grid functionalities

• OGSI
  – Technical specification of the concepts described in OGSA
  – GWI operates on OGSI-based Grid services

• GT3
  – Implementation of OGSI and OGSA
  – GWI operates on GT 3.0
Case Study

- **Obtaining Bayesian Networks form data**
  - Using the unweighted L1 metric spanning tree algorithm
  - Computationally expensive

- **Workflow:**
  1. L1 service reads in data
  2. Unweighted L1 metric measure is computed
  3. Intermediate results are stored
  4. Sort service reads in intermediate results
  5. Intermediate results are sorted
  6. Sorting final results
<workflow name="Simple_Workflow" ...>
  <factoryLinks> ... </factoryLinks>
  <dataLinks> ... </dataLinks>
  <lifecycle>
    <createInstance instance_name="l1_instance">
      <factoryLink name="L1"/>
    </createInstance>
    ...
    <eventHandlers>
      <onNotification instance_name="l1_instance"
        portType="NotificationL1PortType"
        operation="computeL1">
        <destroyInstance instance_name="l1_instance">
          <factoryLink name="L1"/>
        </destroyInstance>
      </onNotification>
    </eventHandlers>

    </lifecycle>
  <controlflow>
    <sequence name="sequence1">
      <invoke instance_name="l1_instance"
        portType="l1_port"
        operation="computeL1"
        dataInFrom="raw_data_input_db"
        dataOutTo="l1_result_db"/>

      ...
    </sequence>
  </controlflow>
</workflow>
Case Study – Activity Diagram

Client

WFE

WFE

L1

L1

Sort

Sort

fact.

create workflow engine instance

return GSH to

the client

factory

parse GWEL
document

create L1
service instance

return GSH to

the workflow

engine instance

instance

store L1

instance GSH

store Sort
instance GSH

invoke L1

instance

invoke Sort
instance

destroy L1

instance

destroy Sort

instance

create Sort

instance

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

send notification

submit GWEL
document

store Sort

service instance GSH

send notification

input data,
compute it and store it

send notification

create workflow

factory

return GSH to

the client

store Sort

instance GSH

invoke Sort

instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
compute it and store it

send notification

create Sort

service instance

create L1

service instance

input data,
Conclusions

• e-Scientists need a generally applicable grid workflow infrastructure
  – Specifying processes graphically
  – Locating available services easily
  – Plugging together services automatically

• Our solution
  – Prototypical infrastructure
  – Feasibility study
  – Introduces core components

• Future work
  – Much more needs to be done
  – Performance issues
  – Implementation of new technologies such as WSRF
Acknowledgements

Most of this work was done at Imperial College London under the supervision of Dr. Steven Newhouse and Dr. Anthony Mayer