Workflow Sharing vs Workflow Interoperability

Gabor Terstyanszky,
University of Westminster
Taverna workshop
30 October 2014
**Formal Description of WEs and WFs**

**formal description of workflows**

\[ WF = \{WF_{abs}, WF_{cnr}, WF_{cnf}, WF_{eng}\} \]

where

- \( WF_{abs} \) - abstract workflow
- \( WF_{cnr} \) - concrete workflow
- \( WF_{cnf} \) - workflow configuration
- \( WF_{eng} \) - workflow engine

**formal description of workflow engines**

\[ WE = \{WE_{abs}, WE_{cnr}, WE_{cnf}\} \]

where

- \( WE_{abs} \) - abstract workflow engine
- \( WE_{cnr} \) - concrete workflow engine
- \( WE_{cnf} \) - workflow engine configuration
Coarse-Grained Interoperability (1)

CGI concept = workflow engine integration

Workflow Engine A → Submission Service Client → Submission Service → Workflow Engine B

Workflow Repository → Distributed Computing Infrastructure
Coarse-Grained Interoperability (2)

- native workflows: J1, J2, J3
- non-native workflows: WF4
  - black boxes which are managed as legacy code applications
CGI Usage Scenario

SHIWA Simulation Platform

- SHIWA Portal
  - Execute native WF
- SHIWA Repository
  - Export WF
  - Import WF
- SHIWA Submission Service

Import WF

Community Gateway

Workflow Engines

DCIs

(workflow developer)

create migrate execute publish

(domain researcher)

import execute WFs

execute WFs

execute WFs
CGI Usage Scenario + Taverna WF

- workflow developer
- Run WF
- Search repo
- Import WF
- Upload WF
- Retrieve WF
- Execute non-native WF

myExperiment Repository

SHIWA Simulation Platform

SHIWA Portal

Execute non-native WF

Upload WF

SHIWA Repository

Retrieve WF

SHIWA Submission Service

Execute non-native WF

Taverna Engine

DCIs

CGI support:
- ASKALON
- Dispel4Py
- Galaxy (???)
- GWES
- Kepler
- MOTEUR
- Pegasus
- PGRADE
- ProActive
- Taverna
- Triana
### Knowledge Transfer and Research Community Support

#### Academic communities

<table>
<thead>
<tr>
<th>Field</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astrophysics</td>
<td>PGRADE + Taverna</td>
</tr>
<tr>
<td>Computational Chemistry</td>
<td>Galaxy + PGRADE + UNICORE</td>
</tr>
<tr>
<td>Heliophysics</td>
<td>PGRADE + Taverna</td>
</tr>
<tr>
<td>Hydrometeorology</td>
<td>PGRADE</td>
</tr>
<tr>
<td>Life Sciences</td>
<td>Galaxy + Moteur + PGRADE + Taverna</td>
</tr>
<tr>
<td>Meteorology</td>
<td>PGRADE</td>
</tr>
<tr>
<td>Material Sciences</td>
<td>PGRADE</td>
</tr>
<tr>
<td>Particle Physics</td>
<td>PGRADE</td>
</tr>
<tr>
<td>Seismology</td>
<td>Dispel4Py + PGRADE</td>
</tr>
</tbody>
</table>

#### Non-academic communities

- Engineering and manufacturing SMEs
  - Business Process Simulation
  - Discrete Event Simulation
  - Fluid Dynamics Simulation
  - PGRADE
Creating and Executing Workflows

workflows in the repository in 2013
abstract - 123
concrete - 119
total - 242

workflows in the repository in 2014
abstract - 213
concrete - 385
total - 598

workflow execution number
SHIWA Simulation Platform
- 512 (dev) / 331 (test) / 181 (training)

Community gateways
Astro workflows - 182 (dev) / 73 (test) 203 (prod)
Compchem workflows - 550 (dyn) / 400 (dock) / 300 (quan)
Helio workflows - 41 (prod)
Life Sciences workflows - 79 (dev) / 325 (prod)
Fine-Grained Interoperability (1)

FGI concept = workflow language translation

Native workflow

IWIR translator

TR1

TR2

TR3

TR4

Front-end

Abstract IWIR workflow

TR1

TR2

TR3

TR4

Concrete JSDL job description

SHIWA DCI BRIDGE

Destination resource

T1

T2

T3

T4

Concrete JSDL job descriptions

Concrete JSDL job descriptions

Concrete JSDL job descriptions

Concrete JSDL job descriptions
Fine-Grained Interoperability (2)

- **front-end plug-in**: converts workflow of workflow engine A into IWIR (Intermediate Workflow Interpreter Representation)
- **back-end plug-in**: converts from IWIR into workflow of workflow engine B

FGI support: ASKALON MOTEUR PGRADE Triana
DCI Interoperability

jobs in non-JSDL

Workflow Engine

J1
J2
J3
J4

JSDL Translator

jobs in JSDL

J1
J2
J3
J4

DCI Bridge

Metabroker

Proxy Server

DCI