Multi-Workflow Systems and Editors

Sandra Gesing
sandra.gesing@uni-tuebingen.de
19 June 2013
Workflow Systems

- Different workflow concepts
- Different workflow languages
- Different workflow constructs

[Images of workflow systems and editors: OGSA-DAI, gUSE, TRIANA, Kepler, Galaxy, Taverna, UNICORE, Workflow 4Ever, KNIME]
Workflow Editors

• Different technologies (workbenches, web-based)
• Different look-and-feel
Heterogeneous Communities

- Diverse research areas
- Diverse workflow engines and editors established

Logical workflows are often the same inside a community
Heterogeneous Communities

- Diverse research areas
- Diverse workflow engines established

Logical workflows are often the same inside a community.
Heterogeneous Communities

• Diverse research areas
• Diverse workflow engines established

Logical workflows are often the same inside a community

⇒ Re-usability of workflows needed for communities
⇒ Workflow Interoperability
Workflow Engine-based Approach

Coarse-grained workflow interoperability

Workflow editor A -> WF A -> Workflow system A -> DCI A

Workflow editor B -> WF B -> Workflow system B -> DCI B
Workflow Engine-based Approach

Coarse-grained workflow interoperability

Workflow editor A

Workflow system A

Workflow editor B

Workflow system B

WF A

Black Box

WF B

Black Box

WF B

DCI A

DCI B

Workflow

Multi-Workflow Systems and Editors
Workflow Engine-based Approach

Fine-grained workflow interoperability

Workflow editor A → WF A → Workflow system A → DCI A

Workflow editor B → WF B → Workflow system B → DCI B
Workflow Engine-based Approach

Fine-grained workflow interoperability

Workflow editor A  ---  Workflow system A  ---  DCI A

Convert to type A

Workflow editor B  ---  Workflow system B  ---  DCI B
Workflow Engine-based Approach

- SHIWA based on gUSE supporting, e.g., Triana, Taverna, Kepler, MOTEUR
- Tavaxy based on Galaxy supporting Galaxy and Taverna
- MoSGrid based on gUSE supporting Galaxy to gUSE, UNICORE in gUSE

Excellent solutions but can be extended...

What happens if a workflow changes?

⇒ Coarse-grained: users need access to the origin workflow editor

⇒ Fine-grained: users needs to change the workflow in the origin and in the targeted workflow editor
Workflow Editor-based Approach
Workflow Editor-based Approach

- Generic workflow editor
- Workflow system A
  - WF A
  - WF B
- Workflow system B
- Black Box
  - WF B
- DCI A
- DCI B

Multi-Workflow Systems and Editors
Workflow Editor-based Approach

Generic workflow editor

Workflow system A

Workflow system B

WF A

WF B

DCI A

DCI B
Workflow Editor-based Approach

Generic workflow editor

WF A

Convert to type A

WF B

Workflow system A

DCI A

Workflow system B

DCI B
Workflow Editor-based Approach

GeWWE (Generic Web-based Workflow Editor)

Goal

• One editor for diverse workflow languages
• Visual representation of workflows is the same as in the origin workflow editor
• Easy integration of new workflow languages

⇒ Same look-and-feel for editing diverse workflows
⇒ One platform for the whole life cycle of editing workflows
Life Cycle of Editing Workflows

Ready-to-process workflow
- uk.ac.bgs.example
  - EDIM1: SQL
  - Split: seis-data
  - Transformer: EDIM2

Workflow instance
- DB: SQL
  - Split: stream
  - Transformer: stream

Visual representation
- GeWWE

Logical representation
- Process
  - Split
  - Process
  - Process
Following the Model-View-Controller (MVC) concept

**Model**
- Process
- Connection
- Connector
- Text
- Registry

**View**
- Icon library
- Process
- Connection
- Connector
- Text

**Controller**
Features for the model and the view
GeWWE – Main Features

- Visualisation of workflows
- Selection of workflow language
- Support of meta nodes
- Selection of processes, connections, and connectors
- Parameterisation of processes
- Annotation
- Registry for internal and external resources
  - processes
  - compute and data resources
- Import and export text
Thank you!
sandra.gesing@uni-tuebingen.de