



3rd CoreGRID Workshop in Grid Middlewares

GWorkflowDL: A Multi-Purpose Language for Scientific Workflow Enactment

Simone Pellegrini, Francesco Giacomini

INFN CNAF
Viale Berti Pichat, 6/2
40127 Bologna, Italy

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

Motivation (1 / 2)

- **Problem:** Workflow Management Systems in the Grid environment are:
 - ✓ **Limited** in describing complex processes
 - Mainly based on *Directed Acyclic Graphs* (DAGs), no support for *loops*
 - ✓ **Bound** to specific Grid middlewares
 - Constructs to interact with a particular Grid infrastructure are provided at language level
 - ✓ **Not interoperable**
 - No *standards* for workflow descriptions and no standards for WfMSs interactions.

Motivation (2 / 2)

- In this paper, we introduce a *novel* approach in the design of a WfMS by using:
 - ✓ **High Level Petri Nets** as a modeling formalism
 - Turing-complete, support for most of the workflow patterns
 - Allow conversion from other formalisms: DAGs, Pi-Calculi
 - ✓ Based on the **micro-kernel** design pattern
 - A *lightweight* system core which implements just few kinds of basic *operations*
 - New **functionalities** can be provided dynamically

Our Vision

- **FACT**: Processes can be expressed composing a finite number of simpler (atomic) operations
 - ✓ A workflow can be used in order to describe how these operations should be composed
- **IDEA**: in a WfMS new functionalities can be provided by means of sub-workflows
- The workflow description language can be used in order to support the enactment process

Outline

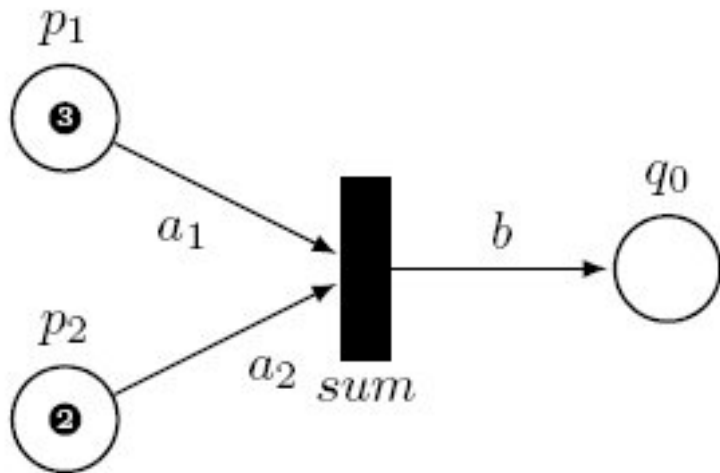
- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

GWorkflowDL (1 / 3)

- Grid Workflow Description Language:
 - ✓ An XML, High Level Petri Net-based workflow description language
 - ✓ Introduced by the FIRST Fraunhofer inside the CoreGRID project
 - ✓ Can be used to represent both **abstract** and **concrete** workflows
- The language has been extended (see [14]) to better address interoperability

GWorkflowDL (2 / 3)

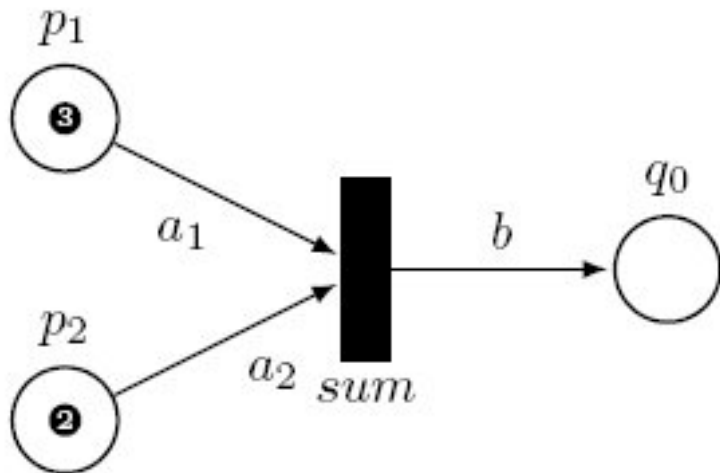
Abstract Workflow



```
<workflow
  xmlns:op="http://www.gridworkflow.org/gworkflowdl/operation">
  <place ID="p1">
    <token><data><t1 xsd:type="xsd:int">3</t1></data></token>
  </place>
  <place ID="p2">
    <token><data><t2 xsd:type="xsd:int">2</t2></data></token>
  </place>
  <place ID="q" />
  <transition ID="T">
    <inputPlace placeID="p1" edgeExpression="a1"/>
    <inputPlace placeID="p2" edgeExpression="a2"/>
    <outputPlace placeID="q" edgeExpression="b"/>
    <op:operation>
      <op:operationClass name="sum"/>
    </op:operation>
  </transition>
</workflow>
```

GWorkflowDL (3 / 3)

Concrete Workflow



```
<workflow
  xmlns:op="http://www.gridworkflow.org/gworkflowdl/operation">
  ...
  <transition ID="T">
    ...
    <op:operation>
      <op:operationClass name="sum">
        <op:wsOperation wsdl="http://localhost/math?wsdl"
          operationName="plus" quality="0.6"/>
        <op:pyOperation operation="b = a1 + a2"
          selected="true" quality="0.3"/>
      </op:operationClass>
    </op:operation>
  </transition>
</workflow>
```

Platform
specific

Extending GWorkflowDL

- In order to make GWorkflowDL compliant with our purposes, a **sub-workflow invocation mechanism** has been defined
- Furthermore, in order to make it *fully* compliant with the HLPN theory:
 - ✓ **Places must have a type** as already specified for tokens (Coloured Petri Nets)
- We also introduced the concept of **time** in order to deal with the *polling* pattern

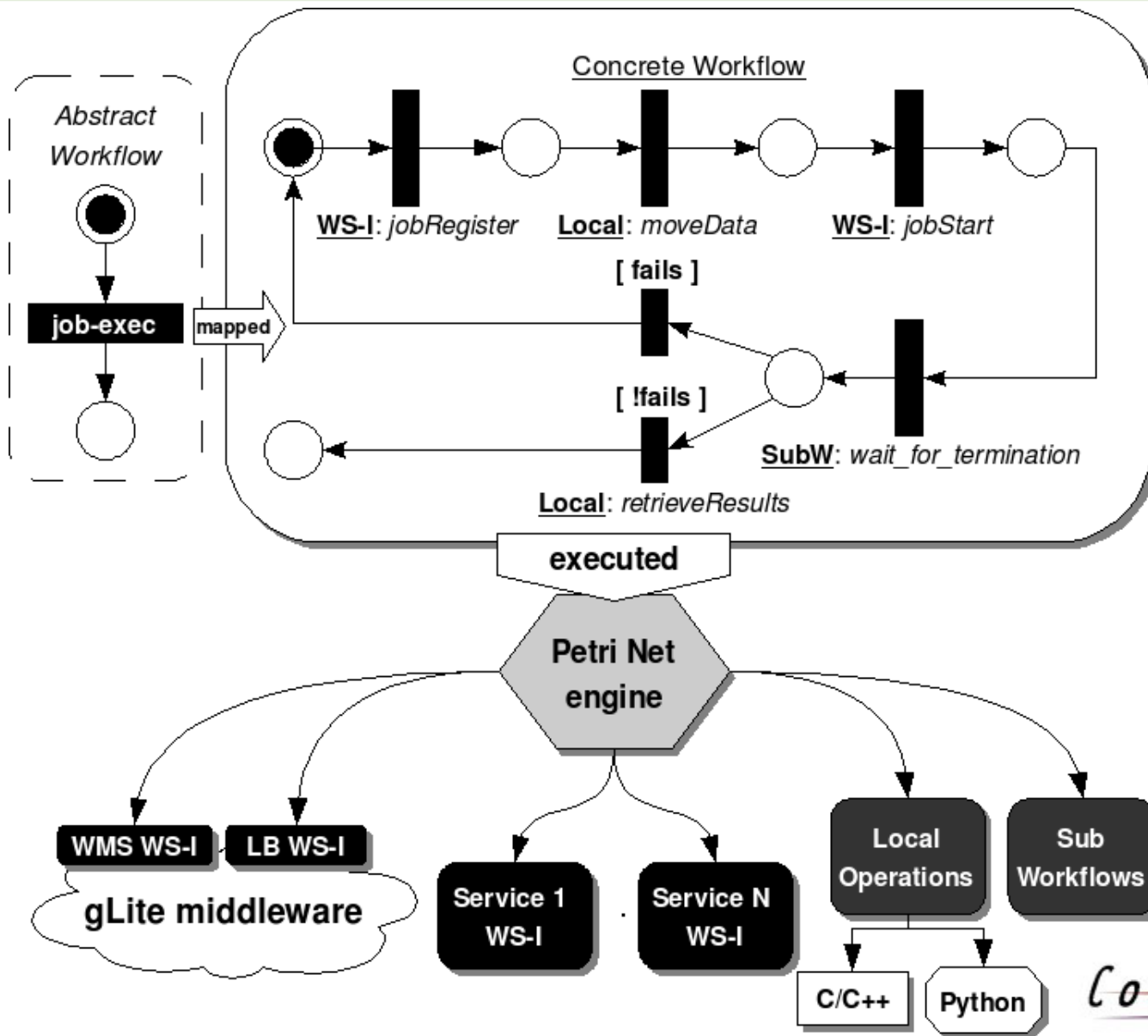
Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

The WfMS Overview (1 / 2)

- The goal is to provide a *lightweight* system core where the workflow **enactment** is achieved by means of GWorkflowDL sub-workflows
- The project focuses in the following aspects of workflow management:
 - ✓ Design of the workflow engine: *fast, reliable and easy to extend*
 - ✓ Conversion between workflow description languages

The WfMS Overview (2/2)



Abstract into Concrete Workflow

- Abstract operations are *mapped* into concrete sub-workflows by the WfMS's **refinement process**
- Currently, the *mapping* is implemented using a simple **associative map**:

`operation_name -> sub-workflow_name`

- ✓ Cannot be considered as a *multi-purpose* strategy
- ✓ Further research efforts are needed (e.g. use of ontologies)

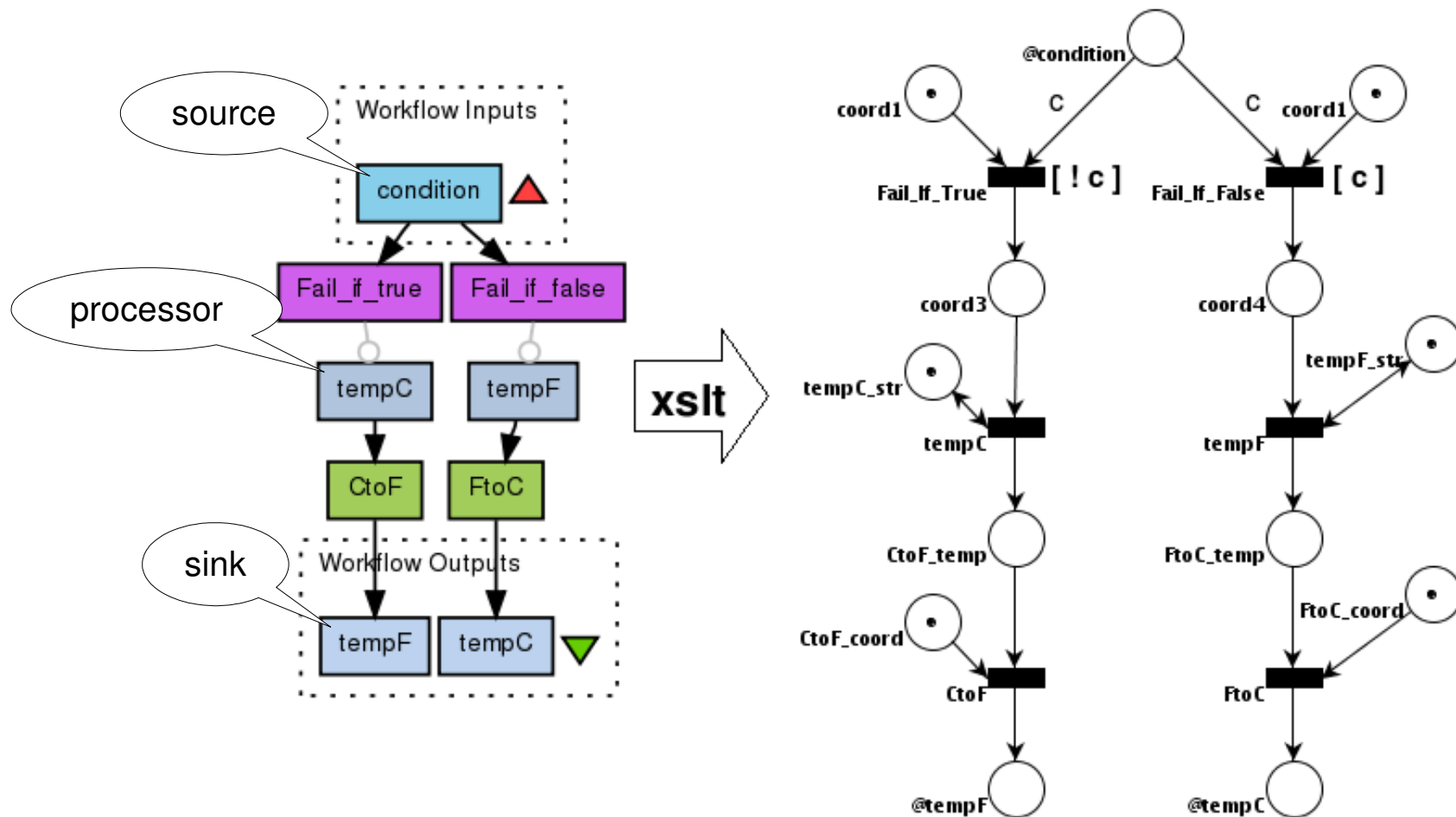
Language Conversion

- The use of the HLPN formalism makes the conversion of existing workflows also possible
- Grid workflows languages (such as the Taverna's SCUFL, DAGMan's ClassAds, etc...) can be converted into GWorkflowDL workflows by **translators**
- **Interoperability** can be obtained allowing the workflow to run **automatically** on different platforms

SCUFL to GWorkflowDL (1 / 3)

- The *Simple Conceptual Unified Flow Language* (SCUFL), is an XML-based workflow language used by the Taverna WfMS
- Main characteristics:
 - ✓ DAG-based
 - ✓ Control structures
 - ✓ Supports sub-workflows
- Conversion to GWorkflowDL is made possible by means of **XSLT**

SCUFL to GWorkflowDL (2 / 3)



SCUFL to GWorkflowDL (3 / 3)

- Taverna is one of the most used WfMS in scientific environment (mainly by bioinformatics)
- However, the SCUFL language cannot be *properly* considered a workflow language:
 - ✓ No concept of abstract workflows
 - ✓ Most of Taverna's workflows have explicitly dependencies with the Java platform
- The conversion of a SCUFL workflow can be performed under certain circumstances

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- **Conclusions and Future Works**

Conclusions

- In this paper we have depicted our contribution inside the CoreGRID project
 - ✓ Introducing a *novel* architecture for a WfMS
 - ✓ Contributing to the definition of a **standard workflow description language** for Scientific Workflows (GWorkflowDL)
 - ✓ Exploring the **language interoperability** by means of translators
- In the future, the WfMS may be integrated in the EGEE/gLite middleware

The End

Questions?



3rd CoreGRID Workshop in Grid Middlewares

GWorkflowDL: A Multi-Purpose Language for Scientific Workflow Enactment

Simone Pellegrini, Francesco Giacomini

INFN CNAF
Viale Berti Pichat, 6/2
40127 Bologna, Italy

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

Motivation (1 / 2)

- **Problem:** Workflow Management Systems in the Grid environment are:
 - ✓ **Limited** in describing complex processes
 - Mainly based on *Directed Acyclic Graphs* (DAGs), no support for *loops*
 - ✓ **Bound** to specific Grid middlewares
 - Constructs to interact with a particular Grid infrastructure are provided at language level
 - ✓ **Not interoperable**
 - No *standards* for workflow descriptions and no standards for WfMSs interactions.

Motivation (2 / 2)

- In this paper, we introduce a *novel* approach in the design of a WfMS by using:
 - ✓ **High Level Petri Nets** as a modeling formalism
 - Turing-complete, support for most of the workflow patterns
 - Allow conversion from other formalisms: DAGs, Pi-Calculi
 - ✓ Based on the **micro-kernel** design pattern
 - A *lightweight* system core which implements just few kinds of basic *operations*
 - New **functionalities** can be provided dynamically

Our Vision

- **FACT**: Processes can be expressed composing a finite number of simpler (atomic) operations
 - ✓ A workflow can be used in order to describe how these operations should be composed
- **IDEA**: in a WfMS new functionalities can be provided by means of sub-workflows
- The workflow description language can be used in order to support the enactment process

Outline

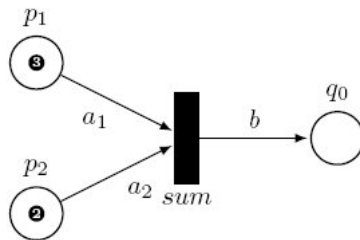
- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

GWorkflowDL (1 / 3)

- Grid Workflow Description Language:
 - ✓ An XML, High Level Petri Net-based workflow description language
 - ✓ Introduced by the FIRST Fraunhofer inside the CoreGRID project
 - ✓ Can be used to represent both **abstract** and **concrete** workflows
- The language has been extended (see [14]) to better address interoperability

GWorkflowDL (2 / 3)

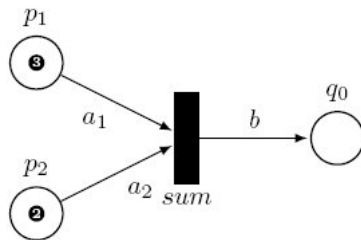
Abstract Workflow



```
<workflow
  xmlns:op="http://www.gridworkflow.org/gworkflowdl/operation">
  <place ID="p1">
    <token><data><t1 xsd:type="xsd:int">3</t1></data></token>
  </place>
  <place ID="p2">
    <token><data><t2 xsd:type="xsd:int">2</t2></data></token>
  </place>
  <place ID="q" />
  <transition ID="T">
    <inputPlace placeID="p1" edgeExpression="a1"/>
    <inputPlace placeID="p2" edgeExpression="a2"/>
    <outputPlace placeID="q" edgeExpression="b"/>
    <op:operation>
      <op:operationClass name="sum"/>
    </op:operation>
  </transition>
</workflow>
```

GWorkflowDL (3 / 3)

Concrete Workflow



```
<workflow
  xmlns:op="http://www.gridworkflow.org/gworkflowdl/operation">
  ...
  <transition ID="T">
    ...
    <op:operation>
      <op:operationClass name="sum">
        <op:wsOperation wsdl="http://localhost/math?wsdl"
          operationName="plus" quality="0.6"/>
        <op:pyOperation operation="b = a1 + a2"
          selected="true" quality="0.3"/>
      </op:operationClass>
    </op:operation>
  </transition>
</workflow>
```

Platform
specific

Extending GWorkflowDL

- In order to make GWorkflowDL compliant with our purposes, a **sub-workflow invocation mechanism** has been defined
- Furthermore, in order to make it *fully* compliant with the HLPN theory:
 - ✓ **Places must have a type** as already specified for tokens (Coloured Petri Nets)
- We also introduced the concept of **time** in order to deal with the *polling* pattern

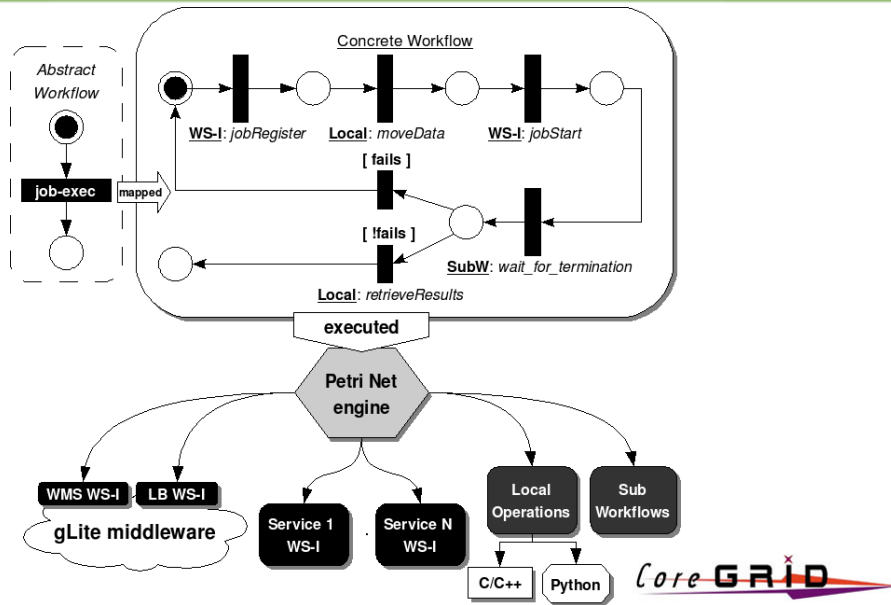
Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

The WfMS Overview (1 / 2)

- The goal is to provide a *lightweight* system core where the workflow **enactment** is achieved by means of GWorkflowDL sub-workflows
- The project focuses in the following aspects of workflow management:
 - ✓ Design of the workflow engine: *fast, reliable and easy to extend*
 - ✓ Conversion between workflow description languages

The WfMS Overview (2 / 2)



Abstract into Concrete Workflow

- Abstract operations are *mapped* into concrete sub-workflows by the WfMS's **refinement process**
- Currently, the *mapping* is implemented using a simple **associative map**:

`operation_name -> sub-workflow_name`

- ✓ Cannot be considered as a *multi-purpose* strategy
- ✓ Further research efforts are needed (e.g. use of ontologies)

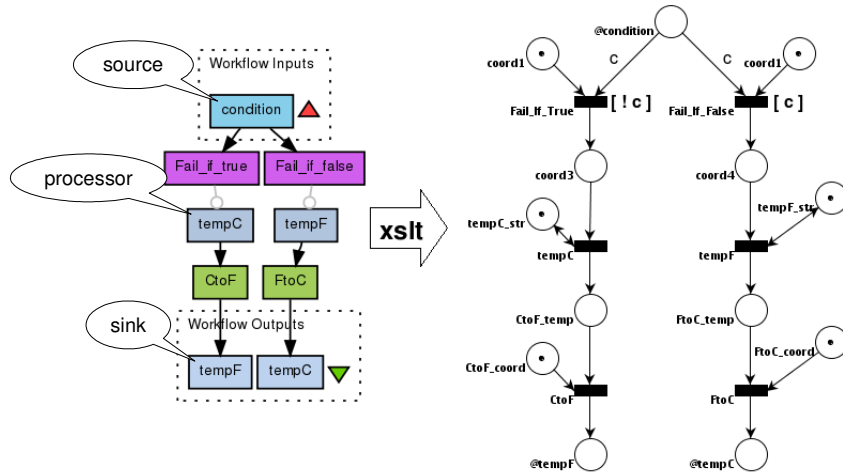
Language Conversion

- The use of the HLPN formalism makes the conversion of existing workflows also possible
- Grid workflows languages (such as the Taverna's SCUFL, DAGMan's ClassAds, etc...) can be converted into GWorkflowDL workflows by **translators**
- **Interoperability** can be obtained allowing the workflow to run **automatically** on different platforms

SCUFL to GWorkflowDL (1 / 3)

- The *Simple Conceptual Unified Flow Language* (SCUFL), is an XML-based workflow language used by the Taverna WfMS
- Main characteristics:
 - ✓ DAG-based
 - ✓ Control structures
 - ✓ Supports sub-workflows
- Conversion to GWorkflowDL is made possible by means of **XSLT**

SCUFL to GWorkflowDL (2 / 3)



SCUFL to GWorkflowDL (3 / 3)

- Taverna is one of the most used WfMS in scientific environment (mainly by bioinformatics)
- However, the SCUFL language cannot be *properly* considered a workflow language:
 - ✓ No concept of abstract workflows
 - ✓ Most of Taverna's workflows have explicitly dependencies with the Java platform
- The conversion of a SCUFL workflow can be performed under certain circumstances

Outline

- Motivation
- GWorkflowDL Language
 - ✓ Extensions
- Workflow Management Systems Overview
 - ✓ Refinement process
 - ✓ Language translators
- Conclusions and Future Works

Conclusions

- In this paper we have depicted our contribution inside the CoreGRID project
 - ✓ Introducing a *novel* architecture for a WfMS
 - ✓ Contributing to the definition of a **standard workflow description language** for Scientific Workflows (GWorkflowDL)
 - ✓ Exploring the **language interoperability** by means of translators
- In the future, the WfMS may be integrated in the EGEE/gLite middleware



The End

Questions?