
# A Business-Oriented Grid Workflow Management System

Luca Clementi, Claudio Cacciari, Maurizio Melato, Roger Menday, and Björn Hagemeier

UNICORE Summit 2007

Rennes, France

European Commission

IST-05-034545

Information Society

# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Introduction 1

- In the last years we have assisted to a wider adoption of Service Oriented Architecture in Grid Computing
- SOA refers to systems structured as network of loosely coupled communication services
- Lately a set of technologies called Web Services have gained a broad acceptance creating a huge collection of:
  - ⇨ Data repositories exposed with WS interfaces
  - ⇨ Application that leverages WS standard for their communication

# Introduction 2

- Therefore for scientists, it is helpful to have instruments that allows them to create simulations that use this existing infrastructure (WS SOA) without having to deal with its intrinsic complexity
- For this reason, the scientific community has developed various workflow systems to orchestrate WS resources. But often they are tailored for the target community so
  - ⇨ Low portability
  - ⇨ Low flexibility

# Outline

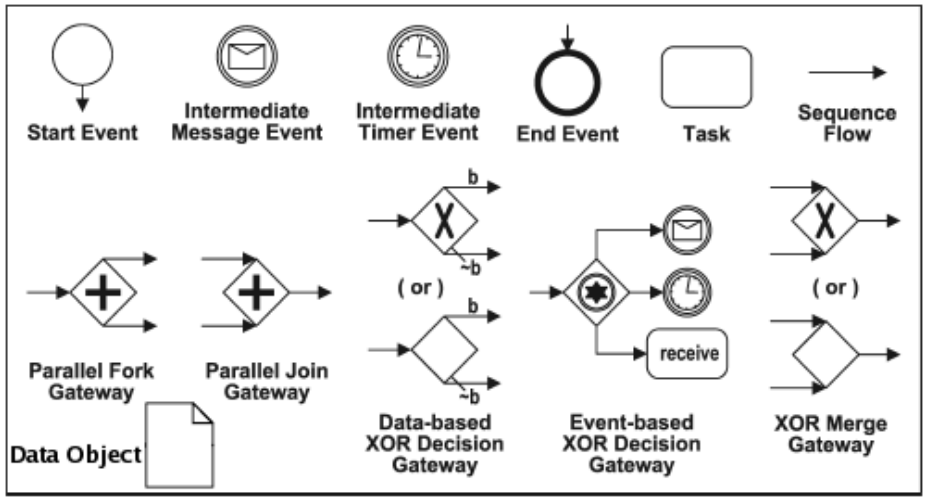
- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Proposed solution - 1

- We believe a more standardized approach should be used to build such a framework
  - ⇨ BPMN and BPEL

# Proposed solution - 2
# BPMN

- Business Process Modeling Notation
- Well standardized graphical notation for business process (aka workflow)
- No standardization for its serialization



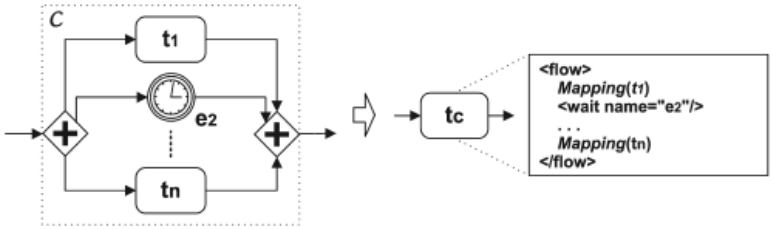
# Proposed solution - 3
# BPEL

- Business Process Execution Language
- Based on WS
- Well standardized and widely adopted by the enterprise community
- The software able to enact a BPEL workflow is usually called ‘engine’
- No graphical representation

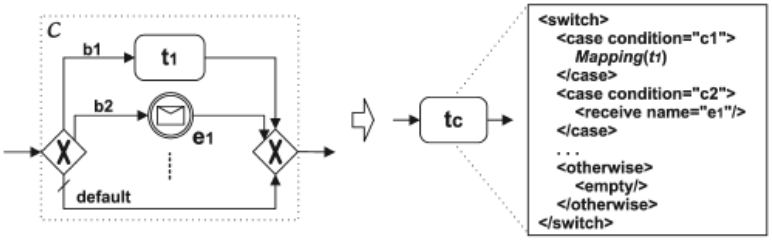
# Proposed solution - 4
# BPMN to BPEL mapping

- **The problem of mapping BPMN and BPEL**

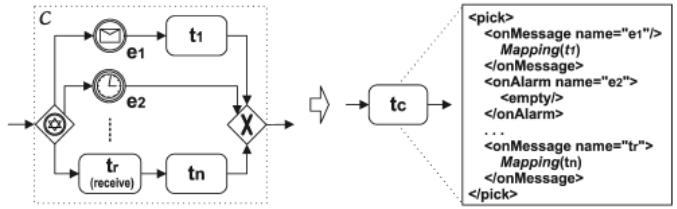

( a ) SEQUENCE-component




( b ) FLOW-component

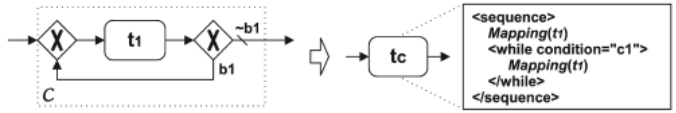



( c ) SWITCH-component

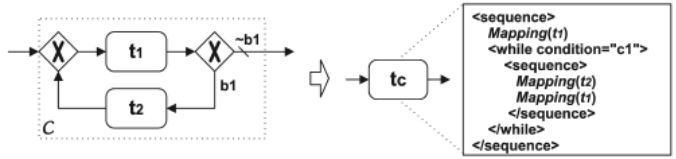



( d ) PICK-component


( e ) WHILE-component




( f ) REPEAT-component




( g ) REPEAT+WHILE-component



European Commission

Information Society

# Proposed solution – 5 BPMN2BPEL

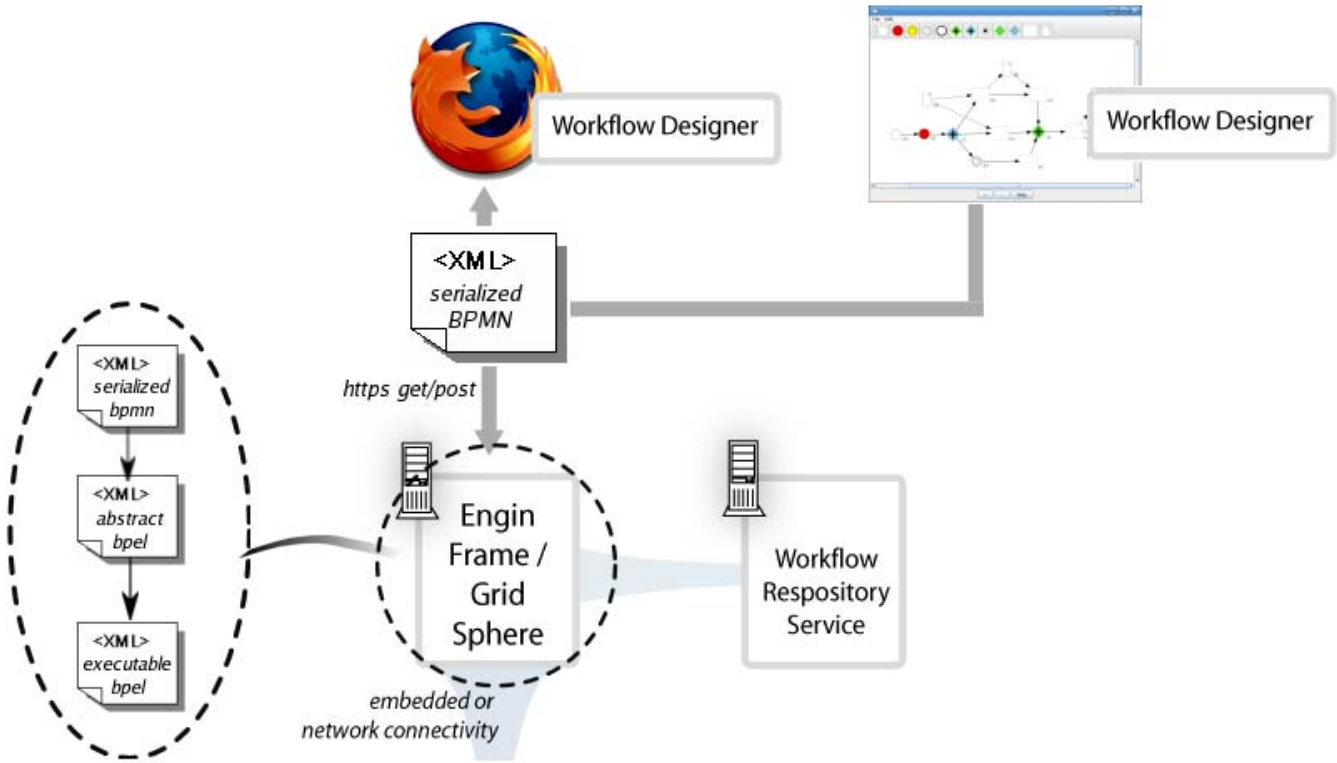
- **A three step process :**
  - ⇨ **The BPMN graph is serialized to an XML document**
  - ⇨ **The XML document is translated into an abstract BPEL document**
    - In an automatic way: through the java library **BPMN2BPEL**.
  - ⇨ **The abstract BPEL is enriched with the pieces of information needed to make it executable (grounding)**

# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Implementation of the solution - 1

- **Workflow design process:**

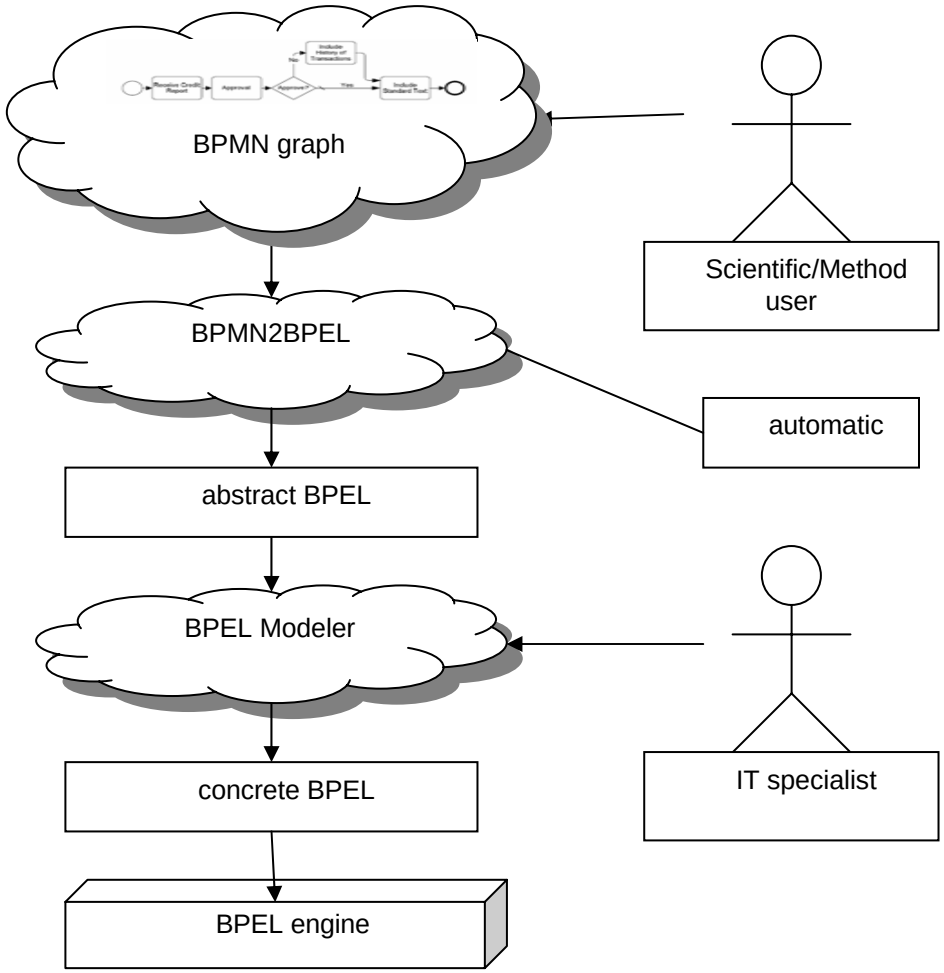





European Commission

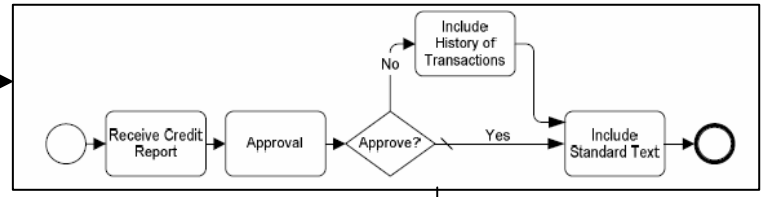
Information Society

# Implementation of the solution – 2 user roles



# Implementation of the solution – 3 user tasks

Scientific/Method Users draw BPMN graph




BPMN2BPEL Automatic Process (challenging!!)

Abstract BPEL can not be executed because it lacks information

BPMN2BPEL library

Abstract BPEL

Grounding:

•Data mapping

•Service Mapping

IT users ground and deploy the BPEL (they know the infrastructure)

Concrete BPEL

Deploy on the Engine

Scientific users invoke the workflow with different parameters

Invoke the workflow

# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Pros e Cons - 1

## Advantage of this approach:

- ⇨ Independence of the workflow description, BPMN, from the workflow executable, BPEL (thanks to the BPMN2BPEL) -> Portability
- ⇨ Method/Scientific users don't care about low level IT details (service mapping, data mapping)

## Disadvantage

- ⇨ How to make data mapping easier (hide XML complexity)
- ⇨ Much more effort: BPMN2BPEL -> three published papers on the algorithm
- ⇨ At present there isn't an implementation of the function BPEL -> BPMN

# Pros e Cons - 2

## BPEL

- **Pros:**
  - ⇨ Several workflow engines available (open source)
  - ⇨ Standard language
  - ⇨ Good integration with web services
  - ⇨ Interactive execution
- **Cons:**
  - ⇨ Needs a deployment phase (not very dynamic)
  - ⇨ No production-level Open Source engines (ODE Apache incubator)*

*Just promoted from incubator to top level Apache project

# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Architecture - 1
# Enterprise Service Bus

- **Enterprise Service Bus is an emerging technology to do integration. From Wikipedia:**

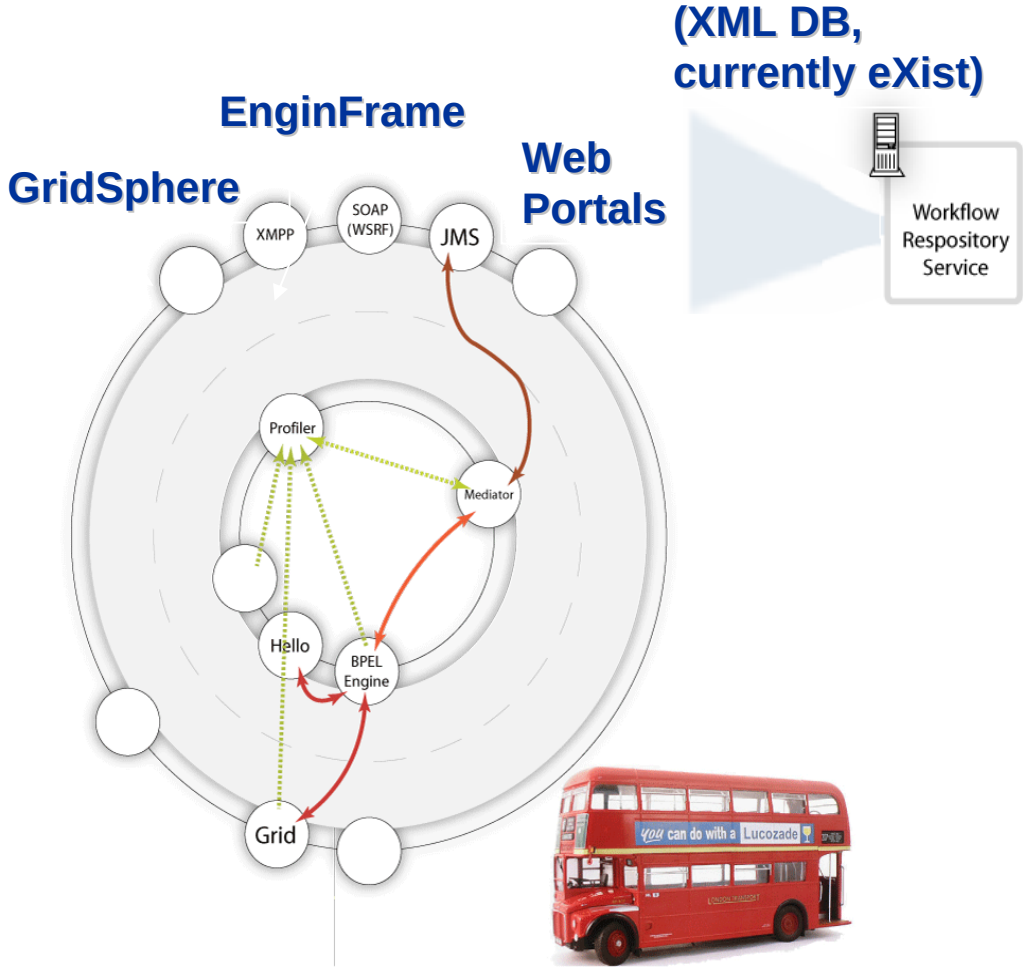
  ***An ESB generally provides an abstraction layer on top of an implementation of an enterprise messaging system which allows integration architects to exploit the value of messaging without writing code.***

- **You can see it as an application container which hosts Components. Components can be divided into:**
  - ⇨ **Binding Components: they are the bridge between external services and the ESB**
  - ⇨ **Service Engines: they provide functionalities to the ESB (business logic)**

# Architecture – 2

- **Aware Service Bus (ASB):**
  - ⇨ EngineFrame, GridSphere and other client options
  - ⇨ Plugable orchestration approaches

**Flexibility**



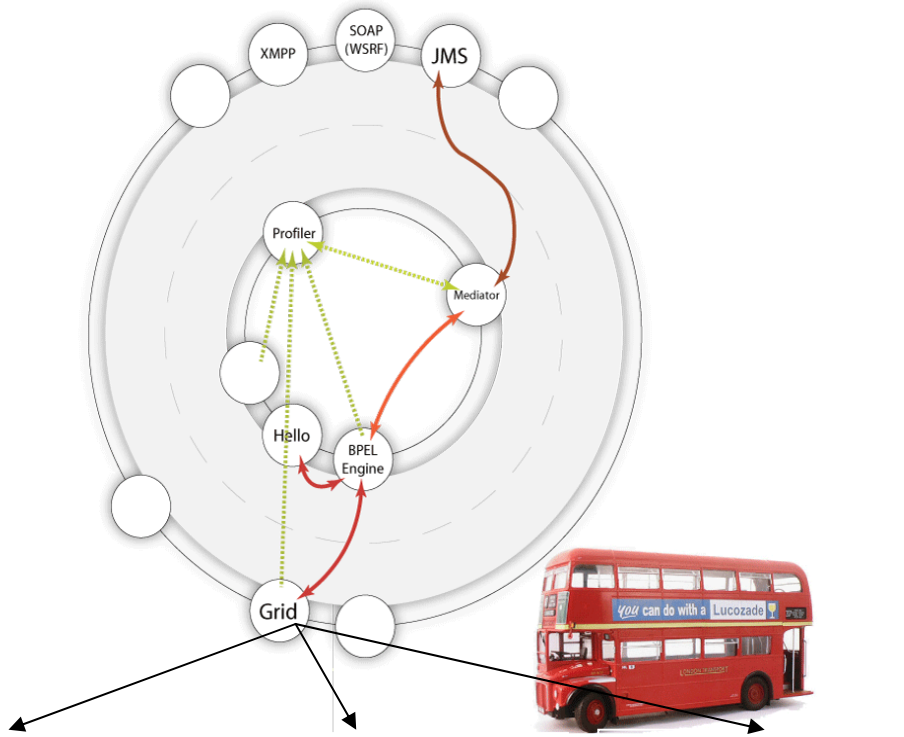



European Commission

Information Society

# Architecture – 3

- **Aware Service Bus (ASB):**
  - ⇨ **Multiple backend Grid infrastructures**



# Outline

- Introduction
- Proposed solution
- Implementation of the solution
- Pros e Cons
- Architecture
- Conclusions

# Conclusions

- The technology developed inside the A-WARE project is able to exploit the resources of the Grid, hiding workflow related complexities.
- A first release of the software is already available on the project web site: http://www.a-ware-project.eu

# Thanks for Your attention !