Towards Uniform BPEL Engine Management in the Cloud

Simon Harrer, Jörg Lenhard and Guido Wirtz  
{firstname.lastname}@uni-bamberg.de  
Distributed Systems Group  
University of Bamberg, Germany

Tammo van Lessen  
tammo.van-lessen@innoq.com  
InnoQ Deutschland GmbH, Germany

Special thanks to Oliver Kopp for fruitful discussions!
BPM lifecycle

- Diagnosis
- Process Design
- Process Enactment
- System Configuration
BPM lifecycle

Status Quo

- Diagnosis
- Process Design
- System Configuration

Process Enactment

Desired Condition

- Diagnosis
- Process Design
- System Configuration

Process Enactment

+ reduce maintenance
+ avoid lock-in & improve portability
+ towards the cloud

Engine Independent
BPM lifecycle

- reduce maintenance
- avoid lock-in & improve portability
- towards the cloud

Engine Independent
BPM lifecycle
for BPEL (and also applicable for BPMN)

Status Quo
- Diagnosis
- Process Design
- System Configuration
- Process Enactment

Desired Condition
- Diagnosis
- Process Design
- System Configuration
- Process Enactment

+ reduce maintenance
+ avoid lock-in & improve portability
+ towards the cloud

Engine Independent
standard conformant BPEL processes
Select engine

standard conformant BPEL processes
standard conformant BPEL processes

Select engine

Install engine
standard conformant BPEL processes

Select engine
Install engine
Start engine
standard conformant BPEL processes

Select engine
Install engine
Start engine
Deploy process
Select engine
Install engine
Start engine
Deploy process
Monitor, Troubleshoot, ... process and its instances
Select engine
Install engine
Start engine
Deploy process
Monitor, Troubleshoot, ... process and its instances

standard conformant BPEL processes

Apache ODE
bpel-g
OpenESB
...
Uniform BPEL Management Layer (UBML)

- Select engine
- Install engine
- Start engine
- Deploy process
- Monitor, Troubleshoot, ... process and its instances

Engine Selection
Engine Provisioning
Engine Lifecycle
Process Deployment
Logfile Access
Process Management
Audit Trail

standard conformant BPEL processes

Apache ODE
bpel-g
OpenESB
...

Select
Install
Start
Deploy
Monitor, Troubleshoot,
process and its instances

Engine
Selection
Engine
Provisioning
Engine
Lifecycle
Process
Deployment
Logfile
Access
Process
Management
Audit
Trail

std:conformant
BPEL
processes

Uniform
BPEL
Management
Layer
(UBML)
Policy of Process Requirements
- Required BPEL features
- Required BPEL extensions
- Required WS-*

Policy of Engine Capabilities
- Supported BPEL features
- Supported BPEL extensions
- Supported WS-*

Match, Rank, Select Best
install & uninstall
- config: port, ...
- dependent software: DB, container, ESB, ...
- environment: hardware, software, ...
start & stop
- to manage resource consumption smartly
deployment archives
- package structure
- deployment descriptors

deployment method
- solely via web service
engine log file(s)
container log file(s)  
... log file(s)
- suspend, rewind & resume
- debug
- manipulate data
- visualize progress
Generic Data Format of Audit Trail
- MXML
Uniform BPEL Management Layer (UBML)

- Select engine
- Install engine
- Start engine
- Deploy process
- Monitor, Troubleshoot, ... process and its instances

- Engine Selection
- Engine Provisioning
- Engine Lifecycle
- Process Deployment
- Logfile Access
- Process Management
- Audit Trail

- Apache ODE
- bpel-g
- OpenESB
- ...
Uniform BPEL Management Layer (UBML)

- Select engine
- Install engine
- Start engine
- Deploy process
- Monitor, Troubleshoot, ...

- Engine Selection
- Engine Provisioning
- Engine Lifecycle
- Process Deployment
- Logfile Access
- Process Management
- Audit Trail

standard conformant BPEL processes

Validated with prototype.

Apache ODE
bpel-g
OpenESB
...

...
Works with **seven** open source engines in various configurations and versions.
1. **Determine the features** in the process
2. **Get engines** supporting the determined features
3. When more than one engine, use the engine with the **higher overall support** for the BPEL specification
Generate Deployment Archive
  Deployment Descriptors
  Package Structure

Use convention over configuration
defaults for endpoint urls.
makeProcessAvailable

1. **select** best engine based on process features
2. **install** engine when not yet installed
3. **start** engine when not yet started
4. **create** deployable archive for process
5. **deploy** deployable archive when not yet deployed
makeProcessUnavailable

1. **undeploy** process when not yet undeployed
2. **stop & uninstall** engine when no process deployed
Limitations (Immediate Future Work)

• Only one engine can be running at the same time due to resource conflicts
  – Solve with configuration options
• Only certain BPEL processes can be converted to deployable packages
  – Generify for arbitrary executable BPEL processes
• Undeploy and isDeployed only implemented for a single Engine
  – Implement this for all BPEL engines
• Prototype only runs on Windows 7 64bit
  – Develop for other platforms (linux)
Process Design

(2) Towards the Cloud
(with LEGO4TOSCA)

(1) Proprietary BPEL Engines
(1) Proprietary BPEL Engines

(2) Towards the Cloud
(with LEGO4TOSCA)

(3) Apply on BPMN Engines
Thank you for your attention!

Any questions?

https://github.com/uniba-dsg/ubml