

Migratability of BPMN 2.0 Process Instances

WESOA 2011

Angineh Barkhordarian Frederik Demuth Kristof Hamann
Minh Hoang Sonja Weichler Sonja Zaplata

Distributed Systems and Information Systems
Department of Informatics
University of Hamburg

05. December 2011



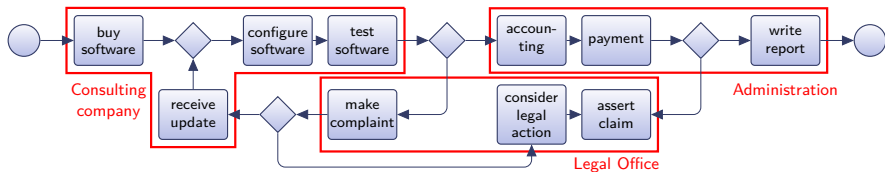
Agenda

- 1 Introduction
 - Why migration?
 - How does it work?
 - A generic solution
 - Why BPMN 2.0?
- 2 Migratability of BPMN 2.0
- 3 Discussion

Why Migrating Process Instances?

Distributed Business Processes

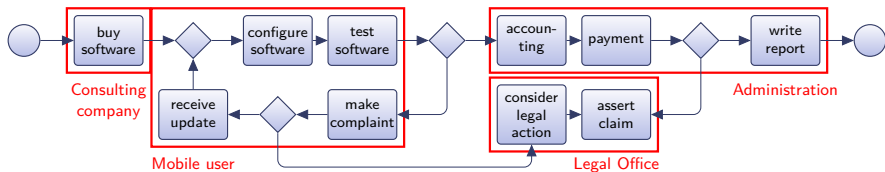
- increasing need for dynamism and flexibility
- e.g. cross-organisational business processes, outsourcing
- **Distribution/decentralisation of resources:**
employees, machines, services, . . .
- **Decentralisation of process execution:**
autonomy of business partners, relevance of location, technological differences, large data transfers, security policy, non-functional aspects, . . .



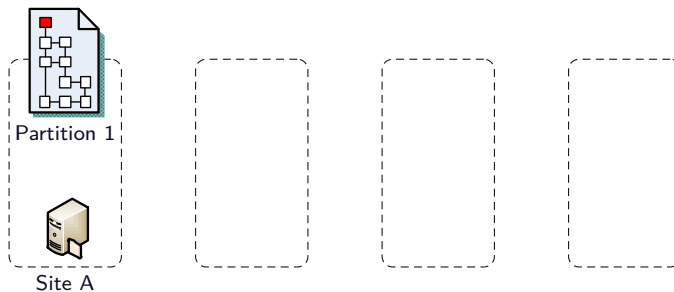
Why Migrating Process Instances?

Distributed Business Processes

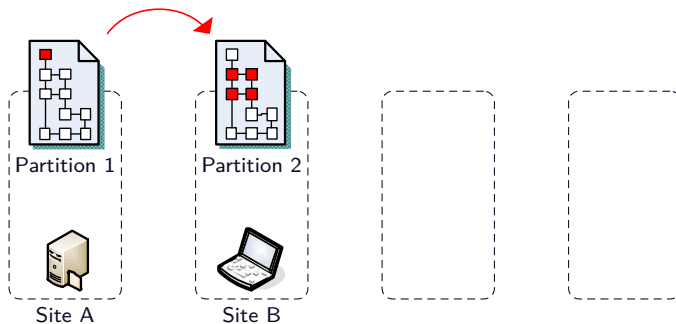
- increasing need for dynamism and flexibility
- e.g. cross-organisational business processes, outsourcing
- **Distribution/decentralisation of resources:**
employees, machines, services, . . .
- **Decentralisation of process execution:**
autonomy of business partners, relevance of location, technological differences, large data transfers, security policy, non-functional aspects, . . .



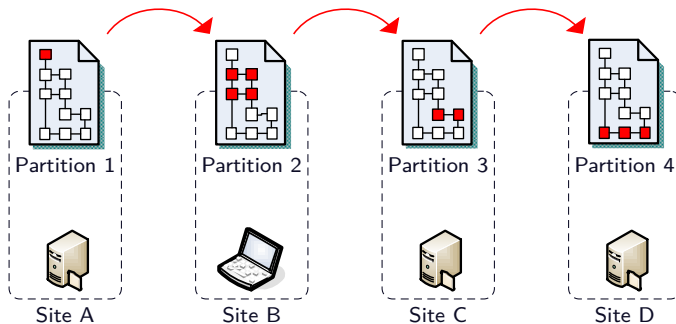
Background: How Does Process Instance Migration Work?



Background: How Does Process Instance Migration Work?



Background: How Does Process Instance Migration Work?



Object of the migration to another process engine are

- 1 the (static) process model (description with all operational aspects)
- 2 state information of the running process instance

Related Work on Process Instance Migration

Research projects using process instance migration (selection)

- introduced by Cichocki and Rusinkiewicz in 1997
- used in OSIRIS, ADEPT Distribution, DEMAC middleware, ...

Problems of these approaches

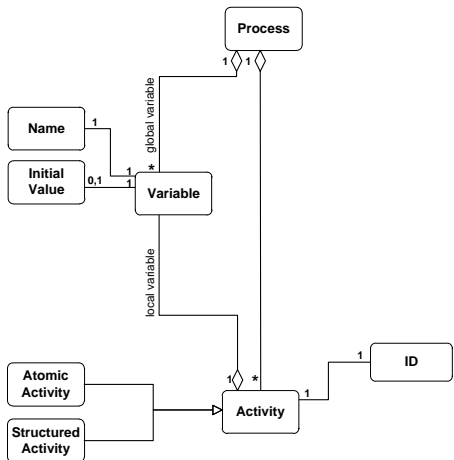
- they introduce either
 - a migration-specific extension of a process description language
 - or a completely proprietary specification
- and therefore require special types of process models

A generic solution

- has to be applicable to arbitrary process description languages
- has to provide a model for missing process instance data

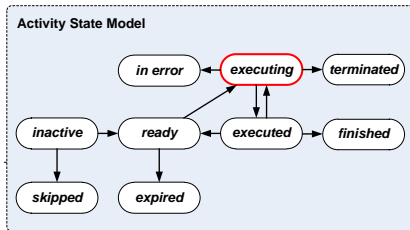
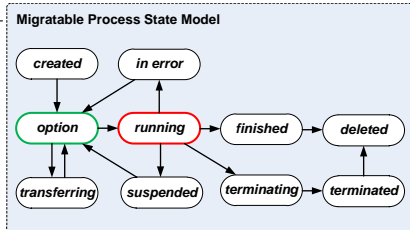
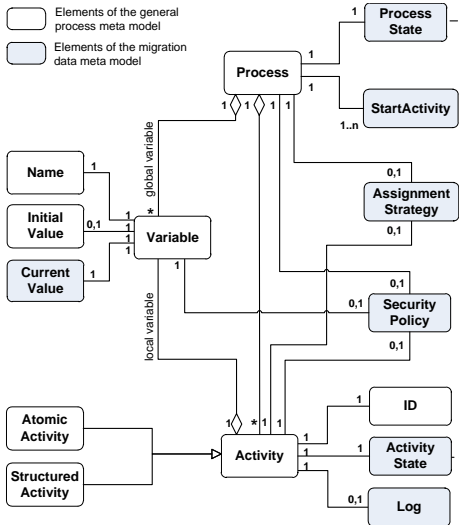
Background: A Generic Migration Meta Model

□ Elements of the general process meta model



Background: A Generic Migration Meta Model

- Elements of the general process meta model
- Elements of the migration data meta model



Why BPMN 2.0?

Opportunities

- promising candidate for enterprises
- one single standard for process analysts and technical developers i.e., it provides both graphical modelling and execution semantic
- *the one standard* in future that will replace EPCs and WS-BPEL?

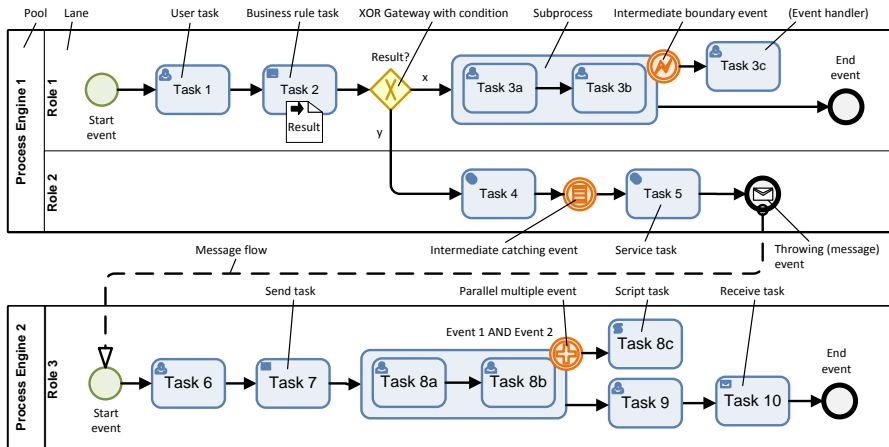
Challenges

- combines interactive user-centric and automatic application tasks
- allows specification on both a specific and abstract level
- provides a strong event concept
- includes constructs for specifying distributed process execution

Agenda

- 1 Introduction
- 2 Migratability of BPMN 2.0
 - Tasks
 - Gateways
 - Sub Processes
 - Events
 - Pools and Swimlanes
- 3 Discussion

Overview: control flow objects in BPMN 2.0



Tasks

- correspond to *atomic activities* in the migration meta model
- if resources are bound dynamically, rebinding is possible after migration – leading to a higher level of flexibility
- if applicable, resources can be moved or copied to the target system



Tasks

- correspond to *atomic activities* in the migration meta model
- if resources are bound dynamically, rebinding is possible after migration – leading to a higher level of flexibility
- if applicable, resources can be moved or copied to the target system

Service Task and Manual Task

- require (local or remote) resources (static or dynamic binding)
- rebinding requires a common understanding e.g. a shared organizational model or an ontology



Service Task



Manual Task

Tasks

- correspond to *atomic activities* in the migration meta model
- if resources are bound dynamically, rebinding is possible after migration – leading to a higher level of flexibility
- if applicable, resources can be moved or copied to the target system

User Task and Script Task

- resource is often deployed with the process (e.g. an HTML form / a small code snippet)
- then, resource can be moved to the target system
- target system has to provide a compatible platform



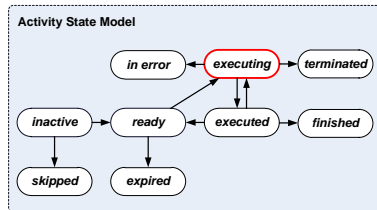
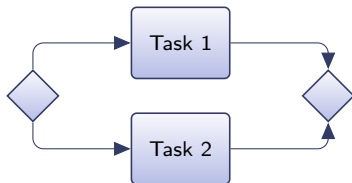
User Task



Script Task

Gateways

- evaluating the gateway conditions as a kind of *atomic activity* which is executed by the process engine itself
- assignment to a specific process engine (or role) is now possible
- activity state model applies



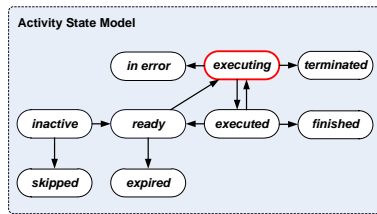
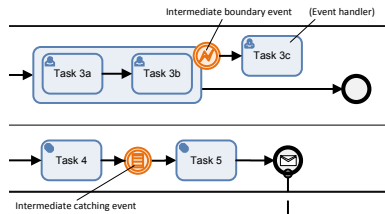
Sub Processes

- correspond to *complex activities* in the migration meta model
- migration is possible even if a sub process is currently executed
- migration can be limited by the state and migratability of enclosed elements
- description of subprocess is required by the target engine; however, it is often included in the same process description

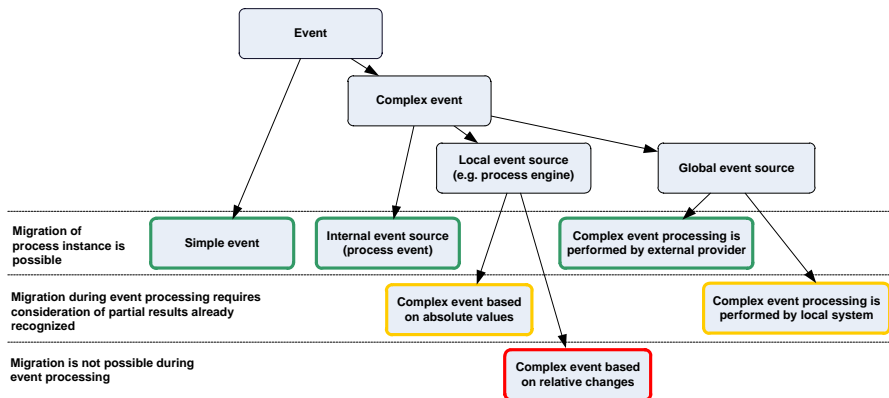


Events

- Throwing an event corresponds to an atomic activity
- Catching an event is more complicated
 - subscription for the event (when? where?)
 - receiving the actual event instance
 - starting the event handler (set of flow objects)
- solution: map catching events to (atomic/structured) activities



Events: a classification



Events: evaluation of specific events in BPMN 2.0

Event name	Purpose	Compl.	Int.	Loc.	Rem.	Migration
Compensation	compensation of activities	-	✓	-	-	possible
Error	hierarchical processing of errors	-	✓	-	-	possible
Escalation	hierarchical processing of escalations	-	✓	-	-	possible
Terminate	terminates the current process instance	-	✓	-	-	possible
Link	connects control flow inside a process	-	✓	-	-	possible
Signal	signaling inside and across processes	-	✓	✓	-	possible ¹
Cancel	canceling transactions	-	✓	✓	-	possible ²
Message	receiving a message	-	-	✓	-	possible ^{1,3}
Timer	wait for a specific time period/point	-	-	✓	-	possible ⁴
Conditional	reference to arbitrary conditions	*	*	*	*	depends on source/complexity
Parallel/Multiple	requires all/one of multiple events	*	*	*	*	depends on used event types

¹ source rebinding possibly required

² see message event in case of external transactions

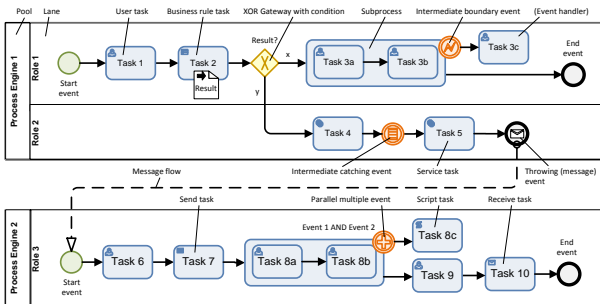
³ the message event is locally triggered on receiving a (remote) message

⁴ time periods can be replaced by an absolute point of time on activation

Pools and Swimlanes

swimlane represents the assignment of flow objects to roles or specific participants (resource level view)

pool determines which unit is responsible for the execution of a specific process partition (assignment strategy)



Agenda

- 1 Introduction
- 2 Migratability of BPMN 2.0
- 3 Discussion
 - Prototype Realization
 - Conclusion

Prototype Realization

Procedure of Process Instance Migration

- ① stop the process engine's execution of a process instance in a consistent state
- ② extract and save runtime information, then delete the process instance or mark it as *transferring*
- ③ send process definition and runtime information to the target system
- ④ resume the transferred process instance on the target system

Experiences

- extension of the open source BPMN 2.0 process engine *ACTIVITI*
- deep intrusions in the engine's code may be necessary
- however, modifications are performed only once (in contrast to physical fragmentation of processes)

Conclusion

Summary

- a meta model for process instance data is not part of BPMN 2.0
- solution: mapping of BPMN elements to a generic migration model
- migration of BPMN 2.0 instances is then – in general – possible
- restrictions: migration is forbidden when tasks (including complex event processing) are currently executed
- resources may have to be copied to the target system

Challenges for Future Work

- BPMN disregards many relevant implementation details (e.g. software bindings, organizational models, user interfaces); respective extensions for BPMN may use additional instance data