Managing Flexibility and Evolution Challenges in Process-aware Information Systems
Scenarios, Technologies, Tools

PROF. DR. MANFRED REICHERT
ULM UNIVERSITY

manfred.reichert@uni-ulm.de
www.uni-ulm.de/dbis

© M. Reichert, 2015
Part 1: Process-aware Information Systems

Part 2: Flexibility Issues

Part 3: Flexibility Support for Pre-specified Process Models

Part 4: Loosely-specified Process Models
Process-Aware Information Systems

A Retail Process

Welcome customer
Offer Clothes
Bill Clothes
Hand over clothes

Mendling 2006
A More Complex Process Scenario

Business Level

Model created by domain expert

IT Level

Model created by IT expert
Business Process Lifecycle
BPM Value Proposition

Value to shareholders and competitiveness

- Transformation
- Business insight
- Compliance & consistency
- IT agility
- Efficiency
- Knowledge

BPM adoption maturity

- Process modeling
- Process execution
- Process monitoring
- Process Optimization

Stakeholders

- Workers, supervisors, and managers
- CIO
- CFO
- CXO
- CEO
- Customers and partners

Forester 2007 BPM Market Overview

© M. Reichert, 2015
Process-aware Information System (PAIS)

<table>
<thead>
<tr>
<th>Late Modeling</th>
<th>Web Clnt API</th>
<th>Modeling API</th>
<th>Dyn. Change API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin. API</td>
<td>Validation</td>
<td>Authorization</td>
<td>Time Mgmt</td>
</tr>
<tr>
<td>Msg Queuing</td>
<td>Exceptions</td>
<td>Audit Trail</td>
<td></td>
</tr>
</tbody>
</table>

Process Execution Engine

- Instance 1
- Instance 2
- Instance 3
- Instance 4
- Instance 5
- Instance 6
- Instance 7
- Instance 8
- Instance 9
- Instance 10
- Instance 11
- Instance 12
- Instance 13
- Instance 14

Process Composer
- Create Process Schema
- Modify Process Schema
- Check Process Schema
- ...

Process Engineer

Process Repository

Application Components

Process Models
A Process Model and Related Instances

Process Schema S

Process Instance I1

Execution Trace: \( \sigma_1 = \langle \text{Patient Admission}, \text{Anamnesis & Clinical Examination}, \text{X-ray} \rangle \)

Activity States:
- Enabled
- Completed
- Skipped
Example

The activities CT and Inform patient are on different branches of an XOR-Block.
Process model to be checked

Generated counterexample:
Execution path and corresponding process context violating the constraint
Levels of Process Model Correctness

Semantic Correctness
(Business Process Compliance)

Behavioral Correctness
(Soundness)

Syntactical Correctness
User Perspective

Process Instance I5

- Patient Admission
- Anamnesis & Clinical Examination
- X-ray
- MRT
- Sonography
- Initial Treatment & Operation Planning
- Operative Treatment
- Discharge & Documentation

Non Operative Therapy
- Non Operative Therapy 1

Offered
- Allocated
- Started
- Completed
- Withdrawn
Let's do the MRT
Process Instance I5

- Patient Admission
- Anamnesis & Clinical Examination

Joe

- Non Operative Therapy
  - X-ray
  - MRT
  - Sonography

Non Operative Therapy 1

Peter

- Initial Treatment & Operation Planning
- Operative Treatment

- Discharge & Documentation

Offered
- Allocated
- Started
- Completed

Withdrawn
User Perspective

Process Instance I5

- Patient Admission ✔
- Anamnesis & Clinical Examination ✔
- Non Operative Therapy
  - X-ray ✔
  - MRT ✔
  - Sonography ✔
- Non Operative Therapy 1
- Initial Treatment & Operation Planning
- Operative Treatment
- Discharge & Documentation

Flow:
- Offered → Allocated → Started → Completed
- Offered → Allocated → Started → Completed
- Offered → Allocated → Started → Completed
- Offered → Allocated → Started → Completed

© M. Reichert, 2015
Flexibility Issues
Processes on the right side of the spectrum are mostly knowledge-intensive

- **Unpredictability**: Course of action depends on situation-specific parameters
- **Non-repeatability**: Two process instances hardly look the same
- **Emergence**: Future course of action depends on knowledge gained through activity execution
Variability is typical for many domains and requires that processes are handled differently depending on the particular context.

- **Drivers**
  - Product and service variability
  - Differences in regulations
  - Different customer groups
  - Temporal differences

Example: Vehicle Repair
Knowledge-intensive processes cannot be fully pre-specified, but require loose specifications

Drivers
- Unpredictability
- Non-Repeatability
- Emergence

Example: Treatment Processes in a Hospital
Ability to adapt the process and its structure to temporary events

Drivers
- Special Situations
- Exceptions

Anticipation of Adaptation
- Planned
- Unanticipated

Example: Examination Procedures in a Hospital
• Ability of the implemented process to change when the business process evolves

• Drivers

External
- Changing Business Context
- Changing Technological Context
- Changing Legal Context
- Organizational Learning

Internal
- Design Errors
- Technical Problems
- Poor Internal Quality

represented in
Real-world Process

provide feedback to
PAIS
Flexibility Issues along the Process Lifecycle

Traditional Process Lifecycle Support

Need for Process Evolution

Process engineer / Process administrator

Execution Log

Process Monitoring

Create Instances

Instance I

Execution Log

Process Execution

Need for Process Adaptation (Support for Planned and Unplanned Exceptions)

Need for Variability Support

Need for Looseness of Process Specifications

[WRW+09]

© M. Reichert, 2015
<table>
<thead>
<tr>
<th>Flexibility Need</th>
<th>Dimension</th>
<th>Technological Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variability</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>Looseness</td>
<td>Loosely-specified processes</td>
<td></td>
</tr>
<tr>
<td>Adaptation</td>
<td>Planned Unplanned</td>
<td>Exception Handling Ad-hoc Changes</td>
</tr>
<tr>
<td>Evolution</td>
<td>Deferred Evolution</td>
<td>Versioning Process Instance Migration</td>
</tr>
<tr>
<td></td>
<td>Immediate Evolution</td>
<td>Refactoring</td>
</tr>
<tr>
<td></td>
<td>Poor Internal Quality</td>
<td>Monitoring, Analysis and Mining</td>
</tr>
<tr>
<td></td>
<td>Organizational Learning</td>
<td></td>
</tr>
</tbody>
</table>

© M. Reichert, 2015
Flexibility-By-Design
Basic Data Flow Concepts & Patterns

Data Edge – Write Access

Data Object

Data Edge – Read Access

Transition Condition references *SparePartsList*

End Message with Data Object *Invoice*
Expressiveness and Flexibility-by-Design

(A)

Preheat Oven → Heat Oil → Add Mushroom Stems → Add Onions → Add Salt → Add Pepper → Cook → Transfer to Bowl along with Spinach, Toss and Cool → Arrange Mushroom Caps → Add Feta to Spinach Mixture → Add Salt to Spinach Mixture → Add Pepper to Spinach Mixture → Divide Filling Between Mushrooms, Bake

Interleaved routing pattern (extended variant)
Variety of related variants

- Same business objective
- Commonalities
- Differences due to varying application context
Configurable Process Models

Main idea: Merging all possible behavior in one reference model with **configurable nodes**
- Extension of an existing process modeling language by adding configurable elements (e.g., activities, control connectors)
- Examples: C-EPC, C-YAWL, Provop

Configurable nodes represent **variation points** associated with configuration alternatives

Possible combinations of configuration alternatives can be restricted through constraints
Configurable Activities

- Included (ON)
- Excluded (OFF)
- Conditional (OPT)
Configurable Control Connectors

- Configurable OR
- Configurable XOR
- Configurable AND

Can be configured to a connector equally restrictive or less restrictive.
Configuration Requirements and Guidelines

- **Requirements**
  - Define constraints over the configuration alternatives that may be chosen

- **Guidelines**
  - Do not prescribe mandatory constraints, but serve as recommendations

![Diagram of Configurable Process Model and Valid Process Variants]
Exception Handling

- Planned
- Exception Handling

- Unplanned
- Ad-hoc Changes
Handling Planned Exceptions in PAIS

I will skip techniques for handling planned exceptions in this lecture and refer to my text book instead!
Handling Unforeseen Exceptions

**Enforcement:**
Guardrails (on a road) prevent deviation, but also prevent anything not predicted.

**Guidance:** Guidelines (on a road) show people where to go, but do not prevent deviations if they are necessary.

K. Swenson, 2014
“Planning is helpful. If you don’t know what you want, you’ll seldom get it. But, no matter how well you plan, you will fare better if you expect the unexpected. The unexpected, by nature, comes unseen, unthought, unenvisioned. All you can do is plan to go unplanned, prepare to be unprepared, make going with the flow part of your agenda, for the most successful among us envision, plan, and prepare, but cast all aside as needed, while those who are unable to go with the flow often suffer, if they survive.”

David W. Jones

I’m sorry Dr. House, I can’t allow you to do that. It would make the process invalid.

K. Swenson, 2014
User View on an Ad-hoc Process Change

Exception — We need an additional lab test!
Behavioral Changes Require Structural Process Model Adaptations
Change Primitives
- Add node
- Remove node
- Add edge
- Remove edge
- ...

High-Level Change Operations
- Combines a set of change primitives
- Referred to as Adaptation Patterns in the following
Structurally Adapting Pre-Specified Process Models

Pattern AP5: SWAP Process Fragment

Description: Two existing process fragments are swapped in process schema S.

Example: Regarding a particular delivery process the order in which requested items are delivered to two customers has to be reversed.

Problem: In a real world process a task has to be accomplished which has not been modeled in the process schema so far.

Design Choices (in addition to those described in Fig. 6):
1. How is the new process fragment X embedded in the process schema?
   a) between two activities (serial insertion)
   b) between two activity sets (parallel insertion with additional condition)

Implementation: This adaptation pattern can be realized by transforming the high level change operation into a sequence of low level change primitives (e.g., add node, add edge).

Related Patterns

Pattern PP3: Late Composition of Process Fragments

Description: At build-time a set of process fragments is defined from which the schema of a concrete process instance can be composed during runtime. This can be achieved by dynamically specifying the control dependencies between them on the fly. Examples of how process fragments can be composed include:

- exam rooms in a hospital
- individual ordering of operations
- automatic generation of instances from a given set of fragments
- building blocks for late modeling
- fragments from the repository can be chosen
- subset of the process fragments from the repository can be s or process fragments can be defined.

Diagram:

- Diagram showing the composition of process fragments with conditions and constraints.
- Example of how fragments X and Y can be composed under certain conditions.

© M. Reichert, 2015
Structurally Adapting Pre-Specified Process Models

Adaptation Patterns vs. Change Primitives

Process Model S

 MOVE ACTIVITY C TO POSITION BETWEEN A AND B

Process Model S'

Snapshot Difference (Change Primitives)

01: Delete edge from A to B
02: Delete edge from B to AND-Split
03: Delete edge from AND-Split to C
04: Delete edge from AND-Split to D
05: Delete edge from C to AND-J
06: Delete edge from D to AND-Join
07: Delete edge from AND-Join to XOR-Split
08: Delete node AND-Split
09: Delete node AND-Join
10: Add edge from A to C
11: Add edge from C to B
12: Add edge from B to D
13: Add edge from D to XOR-Split
Behavioral Changes Must not Violate Process Model Soundness and Proper Instance Execution

Data flow error caused by missing data

No Proper Completion ensured. End node can be reached while B is still enabled
Behavioral Changes Require Adaptations of the Process Instance State

**Dynamic Change Bug**

\[
\begin{array}{c}
\text{a)}
\begin{array}{c}
A \xrightarrow{} B \xrightarrow{} C \xrightarrow{} D
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\text{b)}
\begin{array}{c}
A \quad \text{(Dynamic Change Bug!)} \xrightarrow{} B \xrightarrow{} C \xrightarrow{} D
\end{array}
\end{array}
\]
Behavioral Changes Require Adaptations of the Process Instance State

a) Process Instance I

b) Process Instance I after moving activity C to the position between A and B

c) Process Instance I after placing activity C in parallel to activity B

Activity States
- Completed
- Enabled

© M. Reichert, 2015
Ensuring Dynamic Correctness

May the depicted schema change be propagated to the process instance?

Need for general correctness criterion

⇒ State Compliance

[ReDa98, RRW08a, RRD04a, RRD04b]
Ensuring Dynamic Correctness

Schema $S$:  
\[ \text{activated step} \]

\[ \checkmark \quad \checkmark \quad \checkmark \quad \text{C} \quad \text{E} \quad \text{F} \quad \text{D} \]

Schema $S'$:

\[ \text{make invoice} \]

\[ \checkmark \quad \text{D} \quad \text{E} \quad \text{F} \]

\[ \text{send invoice} \]

\[ \langle \text{A} \rangle, \langle \text{B} \rangle, \langle \text{D} \rangle \quad \Rightarrow \text{Trace reproducible on new schema?} \]

More complicated: loop backs

Further challenges:
- How to efficiently check for compliance?
- How to efficiently migrate process instances?

[RRD04a, RRD04b]
Execution Trace:
\[ \sigma_3 = \langle \text{"Patient Admission"}, \text{"Anamnesis & Clinical Examination"}, \text{"MRT"}, \text{"X-ray"}, \text{"Sonography"} \rangle \]

I3 is not state compliant with change Delete (I3, MRT)

[ReDa98]
Solution for many fundamental research issues!

Formal foundation of the ADEPT technology!
... and its transfer to industrial practice
Flexible Support of Clinical Pathways with AristaFlow

Partners:
Jan Neuhaus, Claudia Reuter
Fraunhoferinstitut Dortmund

Application Example I
Application Example I

Process-aware, Cooperative Emergency Management for Water Infrastructures
Partner: TU Darmstadt

"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change."

Charles Darwin
Process Evolution

Drivers

External
- Changing Business Context
- Changing Technological Context
- Changing Legal Context
- Organizational Learning

Changing Business Context
Changing Technological Context
Changing Legal Context
Organizational Learning

Real-world Process

represented in

PAIS

provide feedback to

Internal

Design Errors
Technical Problems
Poor Internal Quality

© M. Reichert, 2015
process evolution

**Schema S**

- Enter Customer Request
- Check Feasibility
- standard customer → Create Offer
- gold customer → special offer not approved
- special offer approved → Request Approval → Create Special Offer → special offer not approved
- Submit Tender

S evolves to S' by applying change $\Delta_S$ with:

$$\Delta_S = \langle \text{Delete}(S, \text{Create Special Offer}), \text{Delete}(S, \text{Request Approval}) \rangle$$

**Schema S'**

- Enter Customer Request
- Check Feasibility
- Create Offer
- Submit Tender
Change Support Features
Schema Evolution, Version Control and Instance Migration

- **Schema Evolution**
  - Changes at the process type level

- **How to deal with running instances when adapting the original process schema?**
  - Scenario 1: No version control
  - Scenario 2: Co-existence of instances of old / new schema
  - Scenario 3: Change propagation and instance migration
Scenario 1: No Version Control

- Schema is overwritten and instances are migrated

*Type change overwrites schema S*

Process Schema S

Insert X between A and B
Insert Y between C and AND-Join1

Process Instance I1

Process Instance I2

Schema Evolution

Change is propagated to all running process instances

Process Schema S'

Process Instance I1

Process Instance I2

Inconsistent state

© M. Reichert, 2015
Scenario 2: Version Control

- Co-existence of instances of different schema versions

*Type change results into a new version of schema S*

**Old instances remain with schema S**

Instances created from S (before schema evolution)

- **Process Instance I1**
  - **Process Instance I2**

- **Process Instance I4**
  - **Process Instance I5**

**Instances created from S’ (after schema evolution)**

© M. Reichert, 2015
Scenario 3: Instance Migration

- Compliant instances are migrated to the new schema

**Type change results into a new version of schema S**

**Process Schema S**

- Insert X between A and B
- Insert Y between C and AND-Join1

**Process Schema S’**

**Migration of compliant process instances to S’**

**Process Instance I1**

**Process Instance I2**

**Process Instance I2 not compliant with S’**

[RRD04a] © M. Reichert, 2015
Traditional Process Lifecycle Support

1. Create Process Schema
2. Create Instances
3. Process Execution
4. Process Monitoring
5. Evolve Process Schema

Schema S:

Instance I:

Integrated Lifecycle Support for Adaptive and Dynamic Processes (1)
Integrated Lifecycle Support for Adaptive and Dynamic Processes (2)

Lifecycle Support in adaptive PAISs

1. Create Process Schema
2. Create Instances
3. Process Execution
4. Process Monitoring
5. Evolve Process Schema
6. Instance-specific Change
7. Change Propagation

Exception: Delete (I₁, E)

Process engineer / Process administrator

Execution Log

Change Log

Schema S:

A → C → E

B → X → D

Instance I₁:

A → C → E

B → X → D

© M. Reichert, 2015
Loosely Specified Processes

- To deal with unpredictability, non repeatability and emergence loosely specified processes keep (parts) of the process unspecified during build-time.
Late Selection Pattern

**Build-time**

Process Model S

```
A + B + C + D + E + F
```

**Run-time**

Process Instance I₁

```
A + B + C + D + E + F
```

Selected Placeholder Implementation

```
P → Q → R
```

Repository of Potential Placeholder Implementations

- **Fragment 1**
  ```
P → Q → R
  ```

- **Fragment 2**
  ```
  S → T + U → V
  ```

- **Fragment 3**
  ```
  W
  ```

© M. Reichert, 2015
Late Selection – The Worklets Approach

Casuality Treatment Process Model

- Condition
- Conclusion
- Fever = True
  - Treat Fever
- Wound = True
  - Treat Wound
- Abdominal Pain = True
  - Treat Abdominal Pain
- Fracture = True
  - Treat Fracture
- Pregnant = True
  - Treat Labor
- Rash = True
  - Treat Rash
- HeartRate > 190
  - Treat High Heart Rate

[AHE+06]

© M. Reichert, 2015
Late Modeling

Build-time

Process Model S

Run-time

Composed Fragment

© M. Reichert, 2015
Late Modeling – Pockets of Flexibility

Customer Relationship Management Model S

Log Customer Request → Solve Problem → Pocket of Flexibility → Inform Customer

Assign Level 2 Support → Provide Level 2 Support → Document Results

Order Test 912 → Order Test 67 → Order Test 166

Constraints
- Any of the tests may be done, but only one at a time
- Approve Request is required to execute activity Conduct Site Visit

Examples of Sub-Process Fragments Producible for Placeholder Activity

Order Test 912 → Order Test 67 → Order Test 166

Conduct Site Visit

Approve Request
Ad-hoc Composition - Declare

A) Build-time

Activity Templates

- P
- Q
- R
- S
- T
- U
- X
- Y

Constraints

If T no X

B) Run-time

- start(P)
- complete(P)
- start(X)
- complete(X)
- start(Y)
- complete(Y)
Declarative Processes

- Instead of describing exactly how a business process should be executed, declarative processes
  - describe the activities to be executed
  - constraints prohibiting undesired behavior
Declarative Process Model S

Activities A
A  B  C  D  E  F

Constraints C
A  B  C  F

Legend

- **NOT CO-EXISTENCE**
  A and B are mutually exclusive

- **RESPONSE**
  If A is executed, B needs to be executed afterwards

Execution trace producible on S:
\[
\sigma_1 = \langle A, A, D, E, A \rangle \\
\sigma_2 = \langle B, C, F, E, B \rangle \\
\sigma_3 = \langle B, E, F \rangle
\]

Execution trace not producible on S:
\[
\sigma_4 = \langle A, C, E, A \rangle \\
\sigma_5 = \langle B, D, C \rangle \\
\sigma_6 = \langle A, D, B, F, E \rangle
\]
Executing Declarative Processes

Declarative Process Model S

Activities A
- A
- B
- C
- D
- E
- F

Constraints C
- A → B (C1)
- C → F (C2)

NOT CO-EXISTENCE
A and B are mutually exclusive

RESPONSE
If A is executed, B needs to be executed afterwards

Partial Trace | Set of Enabled Activities
---|---
<> | {A, B, C, D, E, F}
<A> | {A, C, D, E, F}
  - B is not included since partial trace <A, B> violates constraint C1
<A, C> | {A, C, D, E, F}
  - B is not included since partial trace <A, B> violates constraint C1

© M. Reichert, 2015
Executing Declarative Processes

<table>
<thead>
<tr>
<th>Activities A</th>
<th>Constraints C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C1</td>
</tr>
<tr>
<td>D</td>
<td>C2</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

Declarative Process Model S

- **Execution**
  - A
  - B
  - C
  - D
  - E
  - F

- **Termination**
  - Activities A, B, C, D, E, F and G are enabled
  - Instance I can terminate, i.e., no termination constraints violated

© M. Reichert, 2015
Executing Declarative Processes

Declarative Process Model S

### Activities A
- A
- B
- C
- D
- E
- F

### Constraints C
- A to B: C1
- C to F: C2

**Timeline**

- Process Instantiation
- A started
- A completed

**Execution**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Termination**

- As A is executed B cannot be executed any longer
- No termination constraint violations, i.e., I can terminate

© M. Reichert, 2015
Executing Declarative Processes

Declarative Process Model S

Actions A
- A
- B
- C
- D
- E
- F

Constraints C
- A
- B
- C
- F

Timeline

Process Instantiation
- A started
- A completed
- C started
- C completed

Execution

Termination

Constraint violations, i.e., I cannot terminate

© M. Reichert, 2015
# Executing Declarative Processes

## Declarative Process Model S

### Activities A
- A
- B
- C
- D
- E
- F

### Constraints C
- A → B
- C → F

## Execution Timeline

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Executing Declarative Processes

Declarative Process Model S

Activities A
- A
- B
- C
- D
- E
- F

Constraints C
- A
- B
- C
- C1
- F
- C2

Timeline

Execution

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Instantiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A started</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C started</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E started</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F started</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F completed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Termination

- No constraint violations, i.e., I can terminate

© M. Reichert, 2015
The Declare System

Composing Declarative Processes with Declare

Executing Declarative Processes with Declare

van der Aalst, Pesic and Schonenberg 2009  [APS09]

© M. Reichert, 2015
Enabling Flexibility in Process-Aware Information Systems
Challenges, Methods, Technologies
Research in my Group

Methods, Concepts and Technologies for Next Generation Process Management Technology

Research Methods
- Design Research
- Formal Methods
- Innovative Prototypes
- Empiric Research

Application Areas
- Healthcare & Psychology
- Automotive Engineering
- Software Engineering
- Product Lifecycle Management
- Sustainable Data
- Mobile Processes
- Business Process Variability
- Data- and Object-Aware Processes
- Human-Centric Processes
- Adapativity & Flexibility

Research Topics
- Research Methods
- Smart Processes
- Business Process Variability
- Sustainable Data
- Mobile Processes
- Healthcare & Psychology
- Automotive Engineering
- Software Engineering
- Product Lifecycle Management
- Sustainable Data
References (Flexibility-by-Design)


References (Configurable Process Models)


References (Exception Handling)


References (Handling Unforeseen Exceptions)


References (Loosely-specified Processes)


[CaSa01] Fabio Casati, Ming-Chien Shan: **Dynamic and adaptive composition of e-services.** Inf. Syst. 26(3): 143-163 (2001)


van Elst; Andreas Lauer; Heiko Maus; Sven Schwarz; Michael Sintek, A.A.A.B.L.: **Frodo: A framework for distributed organizational memories. milestone 1: Requirements analysis and system architecture.** Dfki document (2001). URL http://www.dfki.unikl.de/dfkidok/publications/D/01/01/abstract.html

[Kli00] Justus Klingemann: **Controlled Flexibility in Workflow Management.** CAiSE 2000: 126-141
References (Loosely-specified Processes)


[SuWe03] Hilmar Schuschel, Mathias Weske: Integrated Workflow Planning and Coordination. DEXA 2003: 771-781

References (Declarative Workflows)


