A Decade of Research on "Fluid" Processes: Beyond Rigidity in Business Process Support

Manfred Reichert
Motivation

Permanent new "trends" – require new or adapted services
... which must be integrated

Issues:
- How quickly can processes be implemented?
- At which costs? – With which error risks?
- How expensive will later process changes be?
- How to avoid high maintenance efforts?

Need for Process-awareness
Motivation
Motivation

Schema S:

Create Process Schema

Create Instance

Process designer / Process administrator

Process Monitoring

Execution Log

Process actor

Process Execution

Process Configuration & Implementation

Process Monitoring

Process Execution
Motivation

- Today’s BPM tools are ill equipped to enable **agile enterprises** due to their **inherent brittleness** and **inflexibility**

- Current tool generation implicitly embraces the “**engineer – use** “ **dichotomy** inherited from traditional SE approaches; i.e., systems are first “engineered” and then “used” (or “operated”)

- **Maintenance** and **evolution** are not regarded as part of operation, but rather as interruptions to the “in use” state

- **Role of end users** is not well understood!
Real-world processes are "fluid"!

Why process instances dynamically evolve ...
Why Process Instances Dynamically Evolve …

Example: Transportation Domain

- processes cannot be completely pre-modeled
Why Process Instances Dynamically Evolve …

Example: Healthcare Domain

- Process-aware information systems must not prescribe to physicians how to treat their patients?
Why Process Instances Dynamically Evolve …

Example: Automotive Engineering

- long-running engineering processes cannot be completely pre-modeled

- Example: Release management for E/E-systems in a car
- 200 - 300 control devices to be systematically tested and released
- Requires the execution of hundreds or thousands of process instances
- Concurrent engineering → complex dependencies have to be considered
Why Process Instances Dynamically Evolve …

- Requires to **dissolve the fundamental distinction between “engineering” and “use”;** i.e., end users must be empowered to dynamically evolve processes

- This will lead us to a new class of processes whose “engineering” and “use” is intertwined **fluid processes**

- **Fluid processes are continually adapted and reformed** to **fit the actual needs and constraints** of the situation in hand and to fulfill the overall goals of the involved organizations in the best possible way.
Adaptive Process-Aware Information Systems: Enabling Fluid Processes at Runtime
Adaptive PAIS: Ad-hoc Changes

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

Need for Ad-hoc Changes
Ad-hoc Changes

Exceptional case – we need an additional lab test!

The Users' View

Ad-hoc Changes

Process Type Level

Process Schema S

- Patient Admission
- Anamnesis & Clinical Examination
- X-ray
- Non Operative Therapy
- MRT
- Sonography
- Non Operative Therapy 1
- Discharge & Documentation

Process Instance Level

Process Instance I1

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

Process Instance I2

Execution Trace: $\sigma_2 = \langle \text{Patient Admission} \rangle$
Ad-hoc Changes

For patient „Dijkman“ the MRT activity needs to be skipped due to his cardiac pacemaker.
Ad-hoc Changes

Process Type Level

Process Schema S

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

Ad-hoc Flexibility: Deviations, Change

Process Instance Level

Process Instance I1

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

Process Instance I2

Execution Trace: \( \sigma_2 = \langle \text{Patient Admission} \rangle \)
**Ad-hoc Changes: State Compliance**

**Process Type Level**

**Process Schema S**

- Patient Admission
- Anamnesis & Clinical Examination
- Non Operative Therapy
- X-ray
- MRT
- Sonography
- Discharge & Documentation

**Process Instance Level**

**Process Instance I3**

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

I3 is not compliant with change Delete (I3, MRT)
Ad-hoc Changes: User Assistance

- Annotating changes with information about the reasons for the change
- Retrieval of similar past changes based on context information
- Reuse of changes through PAIS

Process Instance $I_1$: delete($I_1$, MRT)

$pdc_{c_1} = \text{The treatment of cruciate ruptures routinely includes a magnetic resonance tomography (MRT), an X-ray and a sonography. However, for a particular patient the MRT may have to be skipped as the respective patient has a cardiac pacemaker.}$

$qaset_{c_1} = \{(\text{Does the patient have a cardiac pacemaker?}, \text{Patient\.problemList\.hasPacemaker} = \text{Yes})\}$

$sol_{c_1} = \langle\text{Delete}(S, \text{MRT})\rangle$

$freq_{c_1} = 1$
Ad-hoc Changes: The ADEPT Framework

a)

Solution for many fundamental research issues!

b)

Formal foundation of the ADEPT technology!

c)

Ad-hoc Changes: The ADEPT Framework

ADEPT:

Individually adaptable Process Instances
Ad-hoc Changes: The ADEPT Framework

ADEPT:
Individually adaptable Process Instances

Achievements:
- Formal process meta model (expressive + restricted enough)
- Efficient, build-in consistency checks („no bad surprise“)
- Support of a high number of change patterns
- API for accomplishing ad-hoc changes
Adaptive PAIS: Process Schema Evolution
Process Schema Evolution

ADEPT Process Management System

Std Client API  Web Client API  Modeling API  Dynamic Change API
Admin. API  Role Mgmt  Authorization Time Mgmt
Msg Queuing  Recovery  Audit Trail  ...

ADEPT Process Composer
Create Process Template
Modify Process Template
Check Process Template  ...

Repository
Process Templates  Application Components

Process Engine

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

Users

4.377 instances can be automatically migrated
1.117 instances have proceeded too far
123 instances cannot be automatically migrated

Process Designer / Process Administrator

Check Instance Status
The ADEPT Approach
The ADEPT Approach
The ADEPT Approach
Transferring Adept to Practice

AristaFlow BPM Suite

www.aristaflow-forum.de
Adaptive PAIS: Comparing Flexibility Frameworks

- Ability to deal with process changes is among the critical success factors for any process-aware information system (Mutschler et al. 2008)

- Several competing approaches to foster flexibility in process-aware information systems
  - Adaptive workflows (e.g., Reichert & Dadam 1998)
  - Case handling (e.g., van der Aalst et al. 2005)
  - Declarative processes (e.g., Pesic et al. 2007)
  - Late binding / Late Modeling (e.g., Sadiq et al. 2001)

- Lack of methods for a systematic comparison

Adaptive PAIS: Comparing Flexibility Frameworks

Change Patterns

Pattern AP5: SWAP Process Fragment

Pattern AP1: INSERT Process Fragment

Pattern PP3: Late Composition of Process Fragments
Adaptive PAIS: Comparing Flexibility Frameworks

Change Support Features

- Schema Evolution, Version Control and Instance Migration
- Support for Instance-Specific Changes
- Correctness of Changes
- Traceability and Analysis of Changes
- Access Control of Changes
- Change Reuse
- Change Concurrency Control
- Refactoring Support for Process Models
# Adaptive PAIS: Comparing Flexibility Frameworks

<table>
<thead>
<tr>
<th>Primitive / Pattern</th>
<th>Change Primitives</th>
<th>Adaptation Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Academic</td>
<td>Commercial</td>
</tr>
<tr>
<td></td>
<td>ADEPT2 / CBRFlow</td>
<td>CAKE 2</td>
</tr>
<tr>
<td>PR1 – Add Node</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>PR2 – Remove Node</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>PR3 – Add Edge</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>PR4 – Remove Edge</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>PR5 – Move Edge</td>
<td>–</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR1 – Add Node</td>
</tr>
<tr>
<td>PR2 – Remove Node</td>
</tr>
<tr>
<td>PR3 – Add Edge</td>
</tr>
<tr>
<td>PR4 – Remove Edge</td>
</tr>
<tr>
<td>PR5 – Move Edge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adaptation Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1 – Insert Fragment</td>
</tr>
<tr>
<td>AP2 – Delete Fragment</td>
</tr>
<tr>
<td>AP3 – Move Fragment</td>
</tr>
<tr>
<td>AP4 – Replace Fragment</td>
</tr>
<tr>
<td>AP5 – Swap Fragment</td>
</tr>
<tr>
<td>AP6 – Extract Fragment</td>
</tr>
<tr>
<td>AP7 – Inline Fragment</td>
</tr>
<tr>
<td>AP8 – Embed Fragment in</td>
</tr>
<tr>
<td>AP9 – Parallelize Activities</td>
</tr>
<tr>
<td>AP10 – Embed Fragment in Conditional Branch</td>
</tr>
<tr>
<td>AP11 – Add Control Dependency</td>
</tr>
<tr>
<td>AP12 – Remove Control Dependencies</td>
</tr>
<tr>
<td>AP13 – Update Condition</td>
</tr>
<tr>
<td>AP14 – Copy Fragment</td>
</tr>
</tbody>
</table>

Applications

Applying the ADEPT Technology in Practice
Enabling “Fluid Processes” with ADEPT: The Spot Project

Flexible Support of Clinical Pathways with ADEPT

Partners:
Jan Neuhaus, Claudia Reuter
Fraunhoferinstitut Dortmund
Enabling “Fluid Processes” with ADEPT: The Spot Project

Clinical pathways constitute “Fluid Processes” which need to be statically and/or dynamically configured to fit to the patient’s current situation!

- Patient is pregnant; process variant “pregnancy“ is chosen
- New process fragments are inserted in parallel!
- Increased blood sugar level; process variant „Diabetes“ is chosen
Enabling “Fluid Processes” with ADEPT: The Spot Project

The Process Structure Tree - Providing abstraction to end users
Enabling “Fluid Processes” with ADEPT: The Spot Project

The *Process Structure Tree* representing the patient-specific pathway!
Enabling “Fluid Processes” with ADEPT: The Spot Project

Proof-of-Concept Implementation Based on the ADEPT System
Enabling “Fluid Processes” with ADEPT: Disaster Management

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

Semantically Constraining Possible Adaptations in Fluid Processes
How to ensure that process models meet compliance requirements?

How to ensure that flexible ad-hoc process adaptations do not introduce incompliance?

How to trace compliance violations and benefit from past process executions?

Constraint modeling

Process design

Process evaluation

Process execution

Intuitive modeling of semantic constraints using a visual logic-based formalism

Enrichment of process models with semantic constraints

Verification of process models against imposed constraints using graph algorithms and model checking (static verification)

Helpful validation reports and visualization of counterexamples

Runtime verification of dynamic constraints (dynamic verification)

Integrated verification of ad-hoc process changes with regard to compliance

Flexible overriding of constraints if necessary

Documentation of constraint violations and constraint overriding enables traceability and process mining


Ly, Linh Thao and Rinderle, Stefanie and Dadam, Peter (2008) Integration and verification of semantic constraints in adaptive process management systems. Data and Knowledge Engineering, Vol. 64, No. 1, pp. 3-23
After a surgery, an aftercare or an appointment is necessary, before the patient is discharged.

Prior to a CT, the patient has to be informed and after the CT the results have to be reported to the patient.

Prior to an examination of a patient aged beyond 75, an additional tolerance test must be performed.
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
Changes in the Jamuna river (a branch of the Brahmaputra) in Bangladesh between March 1987 (shown in dark blue) and March 1989 (shown in light blue) and superimposed on a SPOT satellite basemap. Change monitoring made it possible to model the river's course and behaviour and to undertake preliminary studies to control flooding.
Monitoring and Mining „Fluid“ Processes

- Create Process Schema
- Evolve Process Schema
- Need for Monitoring and Analyzing Dynamic Processes
- Process engineer / Process administrator
- Process execution
- Change Log
- Execution Log
- Schema S
- Instance I
- Exception: Delete (I, E)
- Change Propagation
- Process participant
- Instance-specific Change
Execution and Change Logs of “Fluid Processes”

Original Schema S

Instance 4711

<table>
<thead>
<tr>
<th>Activity</th>
<th>Event</th>
<th>User</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Started</td>
<td>Garry</td>
<td>2007/09/08 15:00</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Started</td>
<td>Garry</td>
<td>2007/09/08 15:30</td>
</tr>
<tr>
<td>A</td>
<td>Completed</td>
<td>Garry</td>
<td>2007/09/08 15:45</td>
</tr>
<tr>
<td>B</td>
<td>Started</td>
<td>Helen</td>
<td>2007/09/10 11:00</td>
</tr>
<tr>
<td>X</td>
<td>Started</td>
<td>Fritz</td>
<td>2007/09/11 09:01</td>
</tr>
</tbody>
</table>

Change Log Instance 4711 on Schema S

Change TX Applied Changes: User:Timestamp

001 InsertFragment[S;X,A,C]:Helen:2007/09/10 12:02
002 ReplaceFragment(S;C,Z):Jim:2007/09/11 09:31

Process Instance 4711

2007/09/10 11:00

2007/09/10 13:00

2007/09/11 10:00
Change Analysis – A Simple Approach

Change Logs

Fluid processes with instance-specific change logs

Phase I

Change Processes
Change Analysis – A Simple Approach

The discovered meta change process covers all changes applied to at least one of the given fluid process instances.


Change Analysis – A Simple Approach (Proof-of-Concept Prototype)

Change Mining Plugin in ProM

Change Logs Imported from ADEPT
1. Configurations are very costly
2. Variants are difficult to maintain

Derive a new reference process model from the variants such that:

*Less adaptations are needed in future!*

Process Variants Mining: Scenarios

Scenario 1: No original reference process model available

Discovered reference process model $S'$

Scenario 2: Original reference process model known

Process variant $S_1$  
Process variant $S_2$

Customization & adaptation

Process improvement

Original reference process model $S$

Process variant $S_3$  
Process variant $S_4$

Process variant $S_5$  
Process variant $S_n$

Goal: Discover a (new) reference process model which requires less configuration efforts

Process Repository
How to **discover a reference process model**

by mining a collection of **process (instance) variants**

in order to

**reduce the need of future process adaptations?**
Process Variants Mining: Bias and Distance

- **Process Bias**: Minimal set of high-level change operations needed to transform a given process model S into another model S'.

- **Process Distance**: # change operations of any bias between S and S'; can be used to measure the complexity for process change.

---

Process Variant Mining: Reformulated Goal

How to discover a reference process model

by mining a collection of process variants

such that this model has minimum average distance to the process variants?
Process Variant Mining: Illustrating Example

**Process variants:**
- Differ in activity sets
- Differ in process structure and used process patterns
- Differ in relative importance

**Average weighted distance between S and variant models:** 4.85

**Goal:** Can we find a process model which is closer to the variants than S?

Process Variant Mining: Scenarios


Process Variant Mining: Back to our Illustrating Example

S: original reference model

R₄: result after 4 changes (Final result)

R₃: result after 3 changes

R₂: result after 2 changes

R₁: result after 1 change

Δ₂ = delete (R₁, Loop)

Δ₃ = Insert (R₂, X, E, B)

Δ₁ = Move (S, J, B, endFlow)

Δ₄ = Move (R₃, I, D, H)
Process Variant Mining: Evaluation of Search Result

<table>
<thead>
<tr>
<th></th>
<th>S (reference model)</th>
<th>R₁ (After 1 change)</th>
<th>R₂ (After 2 changes)</th>
<th>R₃ (After 3 changes)</th>
<th>R₄ (After 4 changes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness</td>
<td>0.543</td>
<td>0.687</td>
<td>0.805</td>
<td>0.844</td>
<td>0.859</td>
</tr>
<tr>
<td>Average distance</td>
<td>4.85</td>
<td>3.95</td>
<td>3.25</td>
<td>2.65</td>
<td>2.4</td>
</tr>
<tr>
<td>Change operation</td>
<td>Move</td>
<td>Delete</td>
<td>Insert</td>
<td>Move</td>
<td></td>
</tr>
<tr>
<td>Delta fitness</td>
<td>0.143</td>
<td>0.118</td>
<td>0.039</td>
<td>0.009</td>
<td></td>
</tr>
<tr>
<td>Delta distance</td>
<td>0.9</td>
<td>0.7</td>
<td>0.6</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>

Observation:
1. Discovered reference model is getting better with each search step.
2. Most relevant changes are performed at beginning.
Enabling Data-driven Process Structures with COREPRO
The COREPRO Approach: Motivation

Process structure needs to be adapted when product structure changes!
The COREPRO Approach: Motivation
The COREPRO Approach: Motivation

Corepro: Integrated Support of Data-driven Process Structures
The COREPRO Approach: Proof-of-Concept Implementation
Philharmonic Flows

Object-aware Processes
Data Handling in Existing WfMS

1. Access on data only during the execution of activities
2. Missing or incomplete context information
3. No optional activities
4. No control whether the semantic goals are reached
5. Each process instance is executed in isolation
6. Inadequate asynchronous execution of sub-processes

- Generic functions
- Comprehensive lifecycle support
- Missing data-oriented view
Philharmonic Flows: Object-aware Process Management

**data-oriented view**

- application
- applicant
- CV
- cover letter
- send
- evaluation
- decision
- confirmation
- location
- comment
- ...

**process-oriented view**

application

- CV != null
- c.l. != null
- submit != null
- checked = true
- decision = 'accept'
- decision = 'reject'
- confirm. = true
- proposal != null
- submitted = true

- make reviews
- proposal
- submitted
- make reviews
Philharmonic Flows: Object-aware Process Management

Challenge 1: Integrated View

Challenge 2: Clear Granularity!

Challenge 3: Synchronization

Challenge 4: Data-centered Paradigm

Challenge 5: Flexibility
Summary & Outlook

- Business conditions vary with innovation pressure
- Business objectives vary with business conditions
- Business processes vary with business objectives
- Changing business processes will be a common business process in the future
Summary & Outlook

Flexibility Support in most existing PAIS is like Teenager Sex!!

It’s on everyone’s mind all the time.
Everyone’s talking about it all the time
Everyone’s thinks everyone is doing it.
Almost no one is really doing it.
The few who are doing it:
Do it poorly
Think “sure it will be better the next time”.
Are not practicing it safely

Everyone is bragging about their successes all the time, although very few have actually had any success

Anonymous
Summary & Outlook: Projects
Summary

Research Themes

- Adaptiveness & Flexibility
- Correctness & Robustness
- Data & Processes
- Prozess variants
- Distribution & Mobility
- Smart Processes

Killer Applications

- Healthcare
- Automotive Engineering
- Software Engineering
- Product Lifecycle Management
- Logistics
- ... (ellipsis)

Next Generation Process Management Technology

Research Method

- Design Research
- Formal Methods
- Innovative Prototypes
- Empirical Research
Process Visualization & Abstraction

Personalized Visualization
Process Visualization

Summary: What we achieved in Proviado?

**S1 build view**
aggregate & reduce

**S2 assign symbols**
activity:

**S3 fill up symbols**

**S4 calculate layout**

**S5 adapt style**

- color: P1: P3:
- fonts: act.name: Arial 10pt
  actor: Arial 7pt
Navigating in Complex Business Processes

(a) geographic navigation dimension

(b) semantic navigation dimension

(c) view navigation dimension

"logic-based"

"time-based"
**Updatable Process Model Abstractions (Process Views)**

- **Basic Idea:** Using process views not only for visualization purpose, but also as interface for changing the underlying core process model (CPM)
- Updates of a process view then have to be correctly propagated to its CPM as well as all other views on this CPM
- Necessitates a formal foundation
Ambiguities when propagating view changes to the CPM
**Updatable Process Model Abstractions (Process Views)**

*Updating a CPM and related views after a view update!*

![Diagram](image-url)