From Theory to Practice - A Decade of BPM Research & Development in the Automotive Domain: Challenges, Solutions, Project Examples

Manfred Reichert
The Daimler BPM Round Table
Agenda

- Large Process Models
- Large Process Model Collections
- Large Process Structures
Agenda

- Large Process Models
- Large Process Model Collections
- Large Process Structures
The Challenge:
Dealing with Large Process Models
The Challenge: Dealing with Large Process Models
Dealing with Large Process Models: Need for an Advanced Visualization Framework
The Proviado Project
The Proviado Visualization Framework

abstracting information (process views)
adapting visual appearance (symbols, colors, ...)
adapt display form (diagram, form, table, text, ...)

The Proviado Visualization Framework
Proviado: Process Model Abstraction - Example
Proviado: Process Model Abstraction – Basic Operations (1)

Some Requirements:
- Reduce complexity of (large) process models
- Aggregate or eliminate certain process information in a given application context
- Cover all process perspectives: behavior, data, …
Proviado: Process Model Abstraction – Basic Operations (2)

**Reduction**
- Eliminate activities
- Simplify the resulting schema
- Remove adjacent satellite objects

**Aggregation**
- Aggregate activities
- Aggregate adjacent objects if required
Example:

**ShowMyActivities**
Proviado: Process Model Abstraction – Summary

Proviado ...

- offers a powerful mechanism for creating and visualizing process model abstractions (i.e., process views)
- enables a high degree of flexibility in respect to the artefacts created (based on parameterizable view-building operations)
- considers all process perspectives, e.g., control and data flow, process attributes, process logs
- has a well-defined formal foundation
The Proviado Visualization Framework

abstracting information
(process views)

adapting visual appearance
(symbols, colors, ...)

adapt display form
(diagram, form, table, text, ...)
Proviado: Adjusting the Visual Appearance of Process Models

Visualization templates

1. symbol to be used
2. data to be displayed
3. application context
Proviado: Adjusting the Visual Appearance of Process Models

Creating a process visualization

Evaluate the context rules and fill in attribute values
Proviado: Abstraction + Visual Configuration

Personalized Visualization
The Proviado Visualization Framework

- abstracting information (process views)
- adapting visual appearance (symbols, colors, ...)
- adapt display form (diagram, form, table, text, ...)

The Proviado Visualization Framework: A comprehensive approach to visualizing business processes, integrating abstracting information, adapting visual appearance, and adapting display form to enhance understanding and usability.
Proviado: Supporting Different Display Forms for Process Models
The Proviado Visualization Framework: Achievements

S1 build view
aggregate & reduce
"remove activities of P2"

S2 assign symbols
activity: activity name
document: name
actor: name

S3 fill up symbols
symbol: activity_def
activity name
values: name="A"
state="running"

S4 calculate layout

S5 adapt style
color: P1: P3:
fonts: act.name: Arial 10pt
actor: Arial 7pt
Agenda

- Large Process Models

- Large Process Model Collections

- Large Process Structures
The Challenge: Dealing with Large Process Model Collections
... And a Particular Challenge: Managing Process Variants

a) Standardized Repair Process

b) Variant 1: Simple Problem Repair

  Reception → Diagnosis → Repair → Hand Over

  Diagnosis Duration = 2 → Repair Duration = 2

  Maintenance

  b) Variant 1: Simple Problem Repair

  Reception → Diagnosis → Repair → Hand Over

  Diagnosis Duration = 2 → Repair Duration = 2

  Maintenance

  c) Variant 2: Security Critical Repair

  Reception → Diagnosis → Repair → Final Check → Hand Over

  Maintenance

  c) Variant 2: Security Critical Repair

  Reception → Diagnosis → Repair → Final Check → Hand Over

  Maintenance

  d) Variant 3: Security Critical and Simple Problem Repair

  Reception → Diagnosis → Repair → Final Check → Hand Over

  Maintenance

  d) Variant 3: Security Critical and Simple Problem Repair

  Reception → Diagnosis → Repair → Final Check → Hand Over

  Maintenance

  Activity Attribute = Value

  AND Connector

  OR Connector
… And a Particular Challenge: Managing Process Variants

Standard Process:
- Reception → Diagnosis → Repair → Hand Over
- Maintain

Variant 1:
- Fast Diagnosis
  - Reception → Diagnosis (Shortened) → Repair → Hand Over

Variant 2:
- Security Critical Repair
  - Reception → Repair → Hand Over

Variant 3:
- Fast Diagnosis and Security Critical Repair
  - Reception → Diagnosis → Repair → Final Check → Hand Over

Maintenance
- Maintain

Final Check
... And a Particular Challenge: Managing Process Variants

How Daimler captured the variants of a process family ....

**Conclusion:** Both approaches can be supported by commercial tools but do not enable the transparent and explicit management of process variants.
… And a Particular Challenge: Managing Process Variants

Context Model

Business Area

Daimler Financial Services
Daimler Trucks
Daimler Buses
Mercedes-Benz Vans
Mercedes-Benz Cars

Brand
Fuso
Smart
Mercedes-Benz

Vehicle Type
Truck
Van
Bus
Car

Problem: Not all value combinations make sense!
... And a Particular Challenge: Managing Process Variants

Context Model

Business Area
- Daimler Financial Services
- Daimler Trucks
- Daimler Buses
- Mercedes-Benz Vans
- Mercedes-Benz Cars

Brand
- Fuso
- Smart
- Mercedes Benz

Vehicle Type
- LKW
- Van
- Bus
- PKW

Variant 1

Variant 2

Variant 3

...
The Provop Project
The Provop Approach for Managing Process Variants

Base process

Variant specific adjustments
  INSERT IF country = Italy

Configured process variant
  DELETE IF brand = Smart
The Provop Approach for Managing Process Variants
The Provop Approach for Managing Process Variants
The Provop Approach for Managing Process Variants

- **Base Process**
  - Operation x
  - Operation y

- **Options**
  - Option 1
  - Option n

- **Process family**
  - Creating and applying options
  - Configuring Variants
  - Process Modeling
  - Workflow Model

- **Context**
  - Create instance of a specific process variant

- **Workflow Model**
  - Employee
  - Manager
  - GM
  - Word
  - Excel
  - End
Agenda

- Large Process Models
- Large Process Model Collections
- Large Process Structures
The Challenge: Dealing with Large and Complex Process Structures
Automotive Engineering:

- Electrical control units (ECUs) become more and more important:
  - provide many safety-critical functions
  - fast implementation of changes: adjustments and bug fixes by flashing new software onto the ECU

- Modern cars comprise up to 70 ECUs; >10,000,000 LoC

- ECUs interconnected by up to 10 buses with 2 kilometers of wires

- 90% of car innovations enabled by E/E systems

Example: Electronics in side door

- **Power window**
  - Safety stop
  - Close with central locking system
  - Safety functions (Presafe)
  - Communication with air condition

- **Electrical side mirrors**
  - Electrical adjustment
  - Electrical heating
  - Memory function
  - Retractable side mirror
  - Automatic fading out
  - Ambient illumination
  - Turn indicator

- **Door lock**
  - Open / Close with central locking system
  - Sensors for alarm system
  - Power closing

- **Sidebags**
  - Side impact sensors

- **Active surround speakers**

- **Control unit for**
  - Power windows
  - Mirror adjustment
  - Seat adjustment
  - Memory function
  - Child safety lock
  - Central locking system
The Challenge: Dealing with Large and Complex Process Structures

Current Problems in Automotive Engineering

- Up to 50% of all car breakdowns due to electrical / electronic problems

- Some facts
  - Many non-obvious dependencies between ECUs
  - Different life and development cycles of mechanics, hardware and software
  - Numerous ECU variants and versions

- Systematic verification and release management required
The Challenge: Dealing with Large and Complex Process Structures
The Corepro Project
The Corepro Project – Basic Approach

**Modellebene**

<table>
<thead>
<tr>
<th>Datenmodell</th>
<th>Datenmodell</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Model</td>
<td>Object Life Cycles / Life Cycle Coordination Model</td>
</tr>
</tbody>
</table>

**Instanzebene**

<table>
<thead>
<tr>
<th>Datenstruktur</th>
<th>Datenstruktur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>Data-driven Process Structure</td>
</tr>
</tbody>
</table>
The Corepro Project – Basic Approach

Object Life Cycles / Life Cycle Coordination Model

Data-driven Process Structure
The Corepro Project – Basic Approach

Data-driven Process Structure
- Significant reduction of modeling efforts for process engineers
- Formal operational semantics allows for correct executability
- Soundness can be guaranteed on an abstracted level
The Corepro Project – Exception Handling

Graphical User Interface of COREPROsim
Automatic Creation and Execution of a Process Structure
Dynamic Adaptation and Exception Handling
Simulating Large Process Structures
Defining the Model Level with Correctness Checks
Concluding Remarks