Large Process Models and Process Model Collections:
- Challenges, Methods, Technologies -

Barbara Weber

Victoria Torres

Manfred Reichert

Presenters

Barbara Weber
Barbara.Weber@uibk.ac.at
University of Innsbruck

QUALITY ENGINEERING

Victoria Torres
vtorres@pros.upv.es
Polytechnic University of Valencia

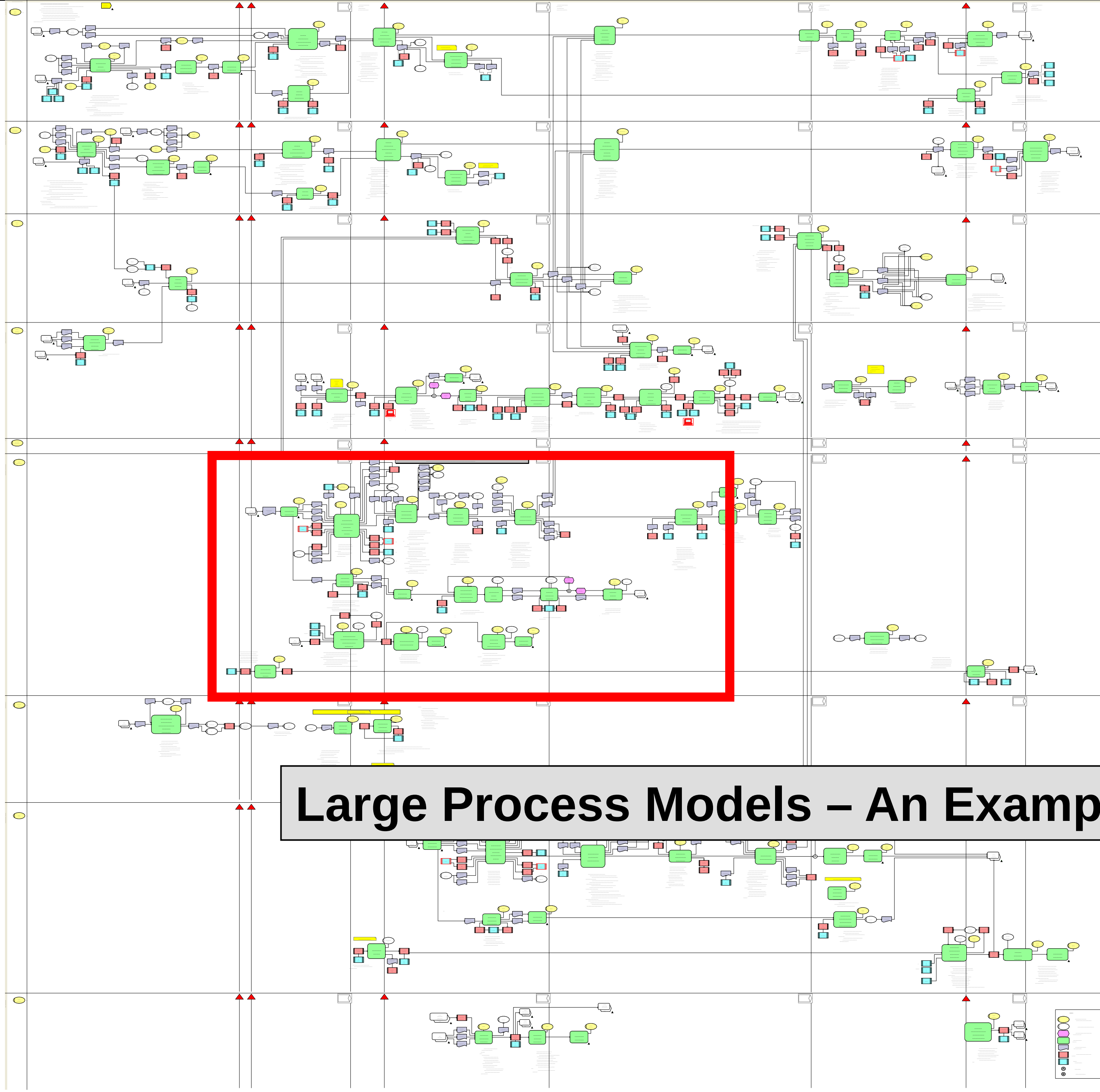
PROS
Centro de Investigación en Métodos de Producción de Software

Manfred Reichert
manfred.reichert@uni-ulm.de
University of Ulm

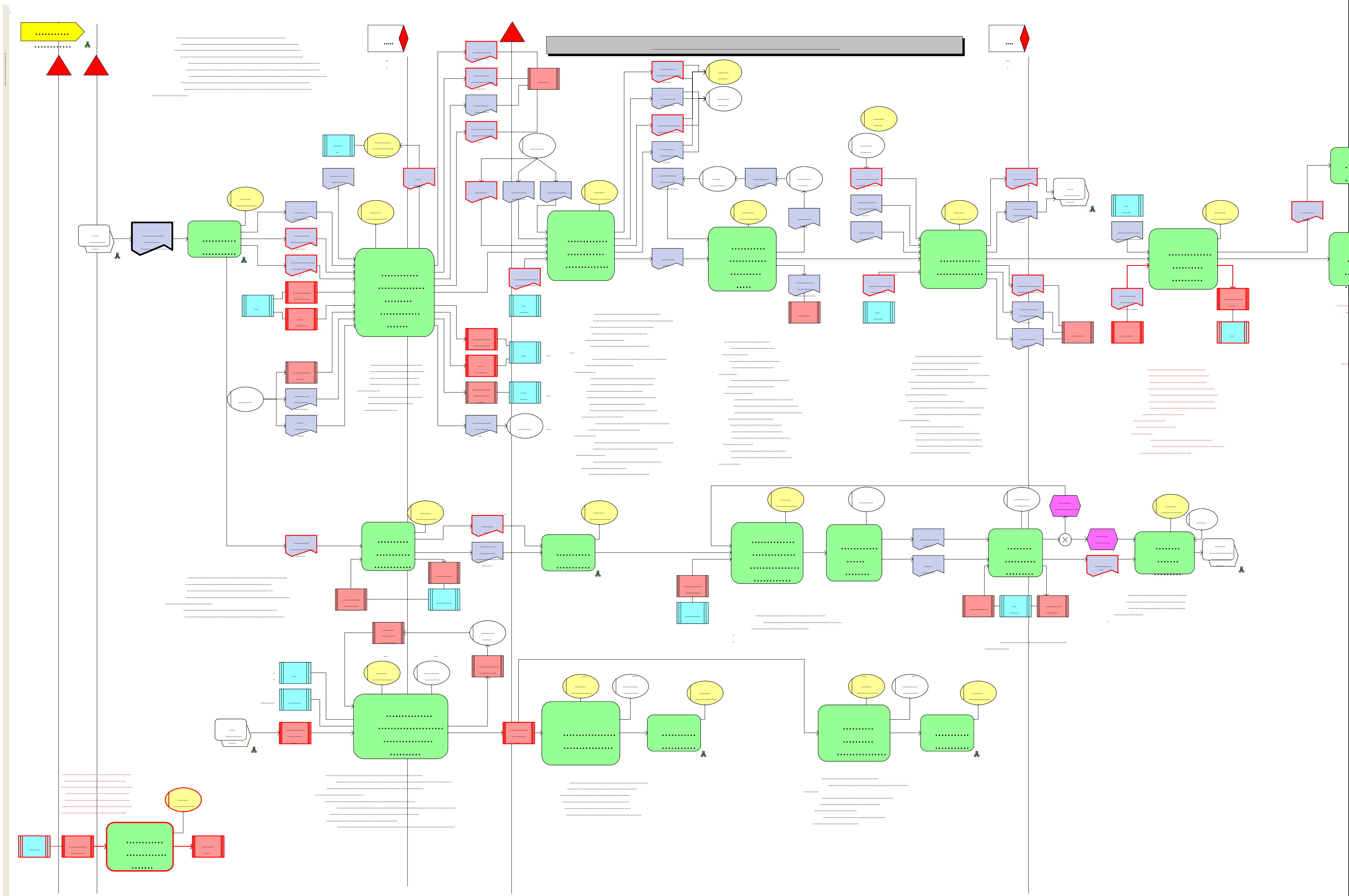
DBIS EXPERIENCE
processes for a flexible world

Agenda

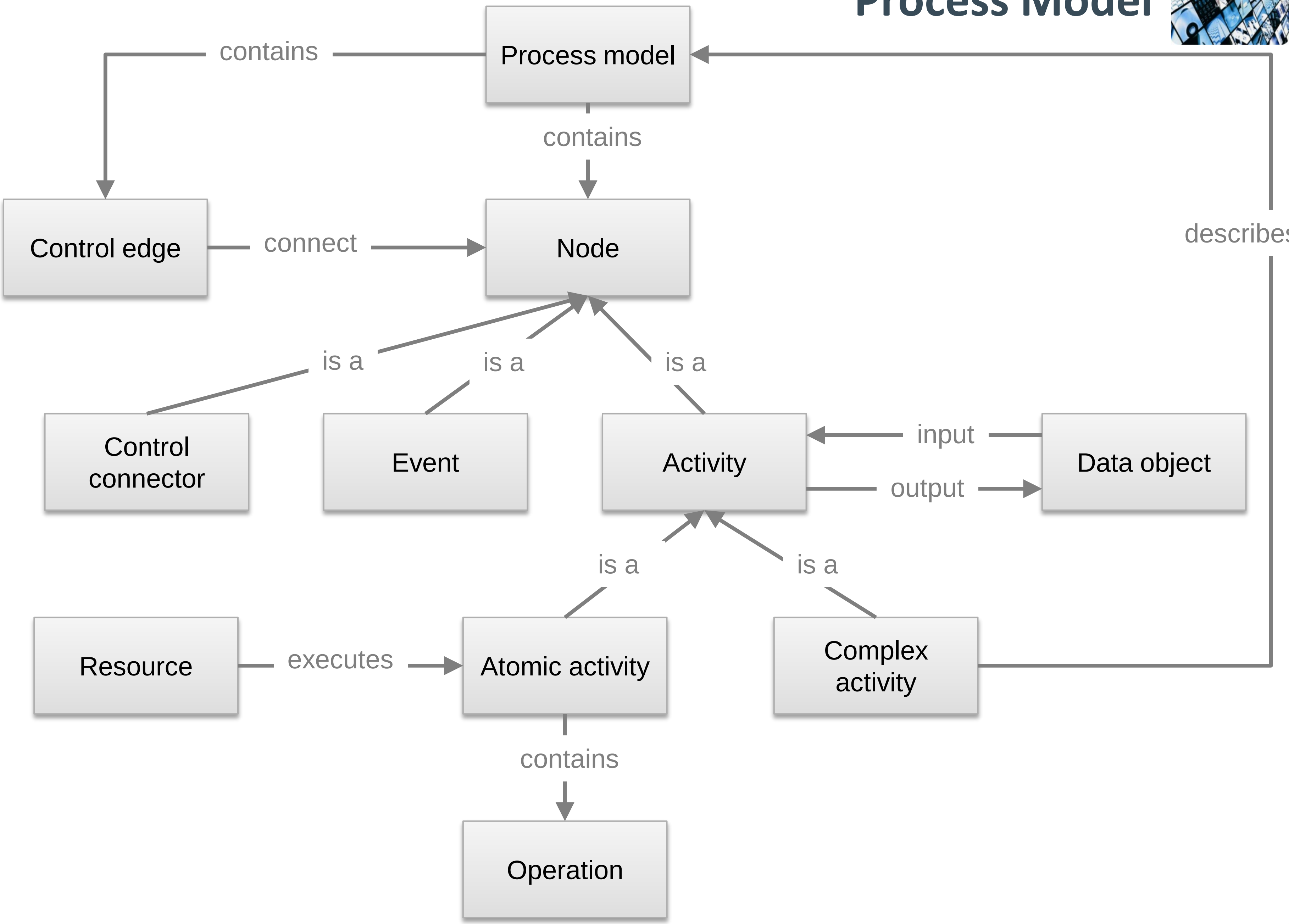
- Challenges & Basic Notions
- Part I: Large Process Models
- Part II: Large Process Model Collections
- Part III: Large Process Structures
- References



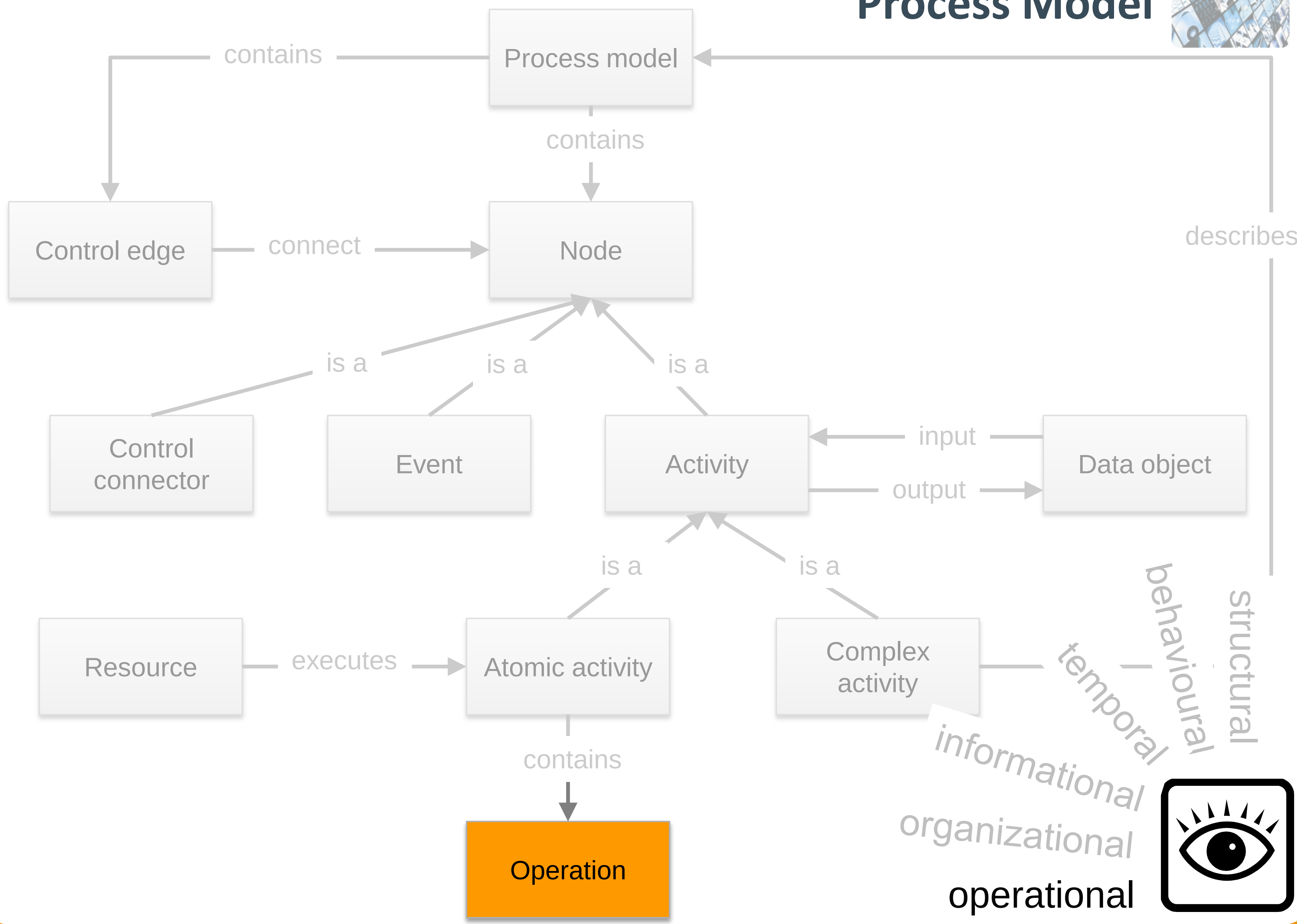
Large Process Models – An Example!



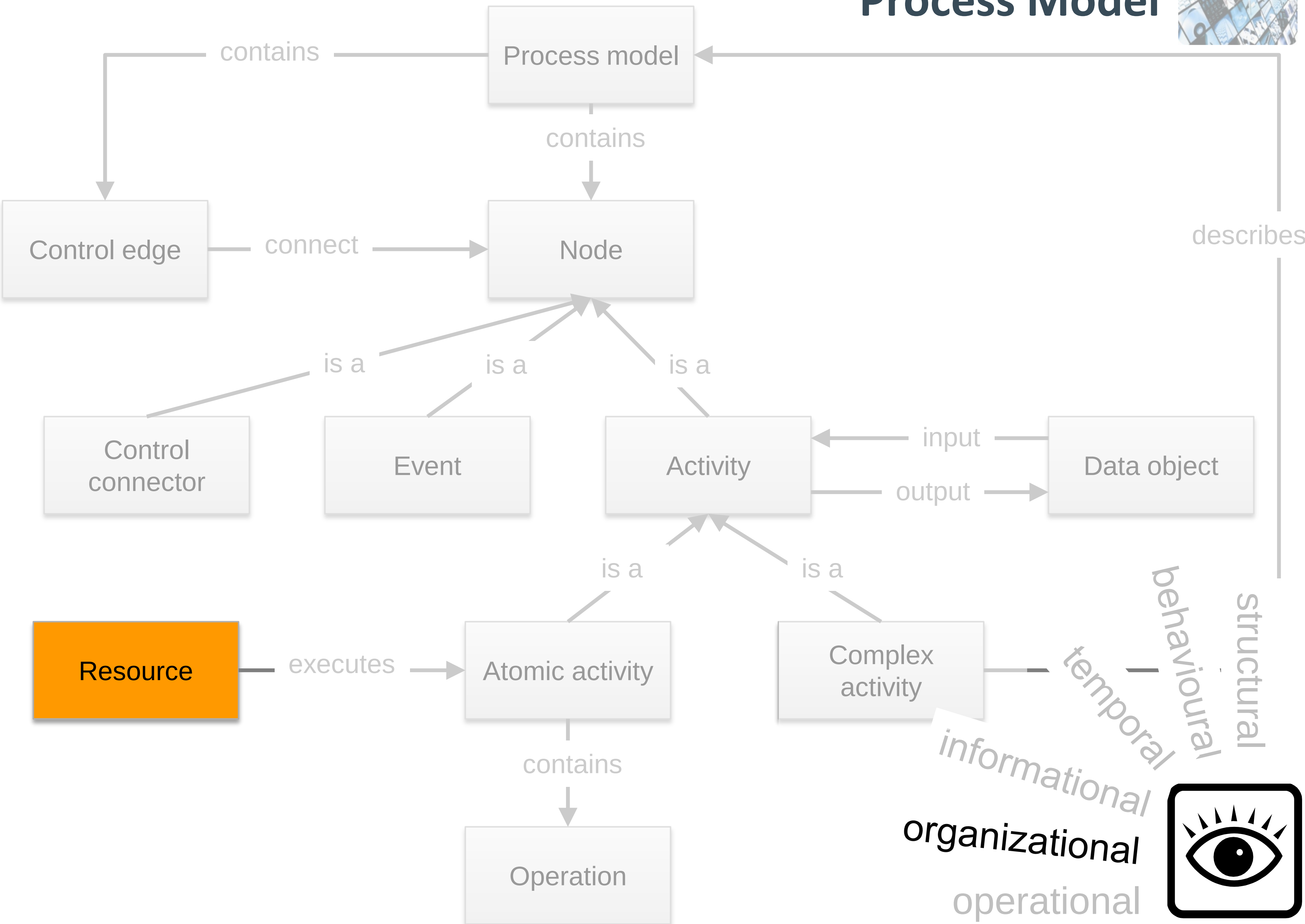
Process Model



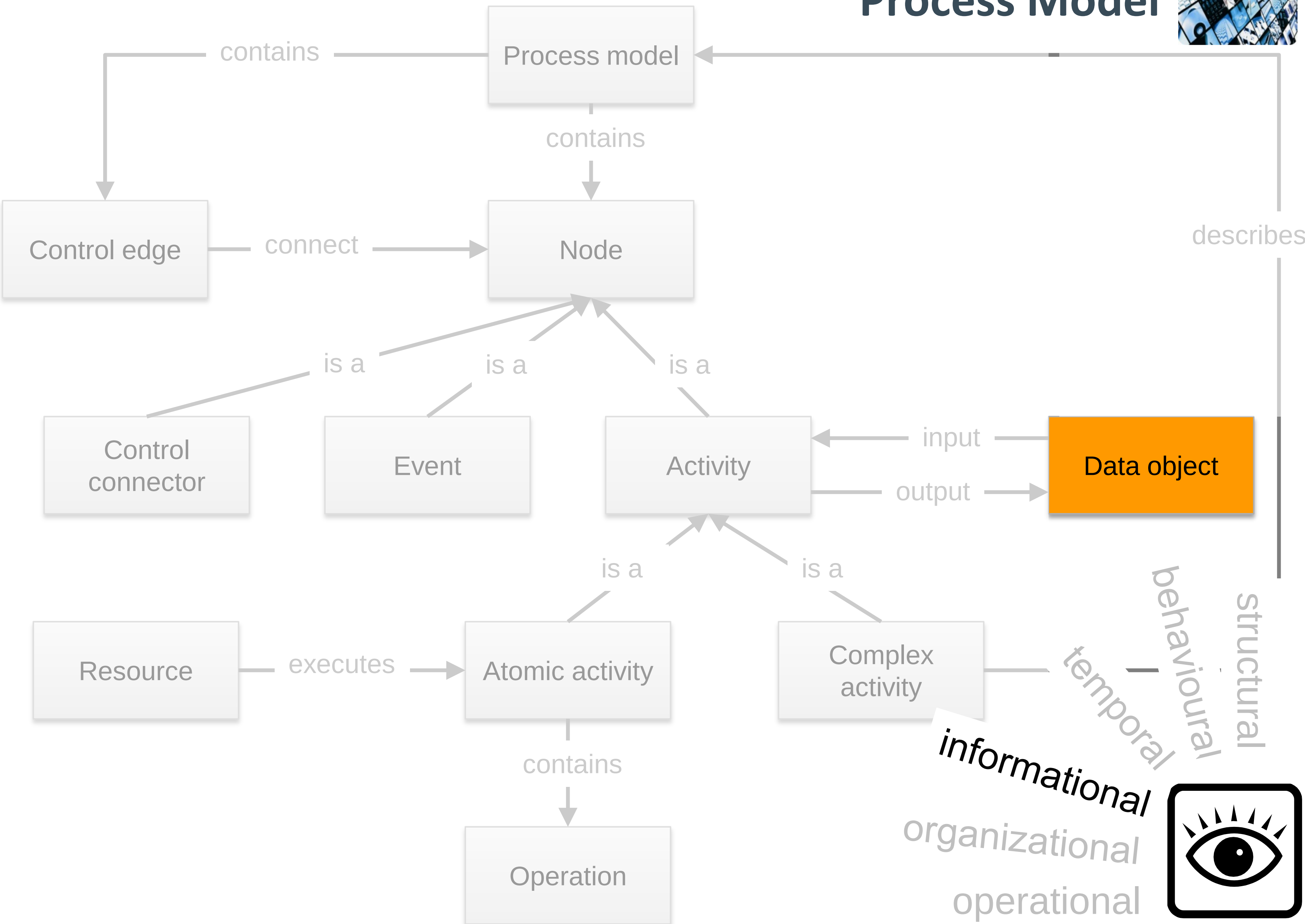
Process Model



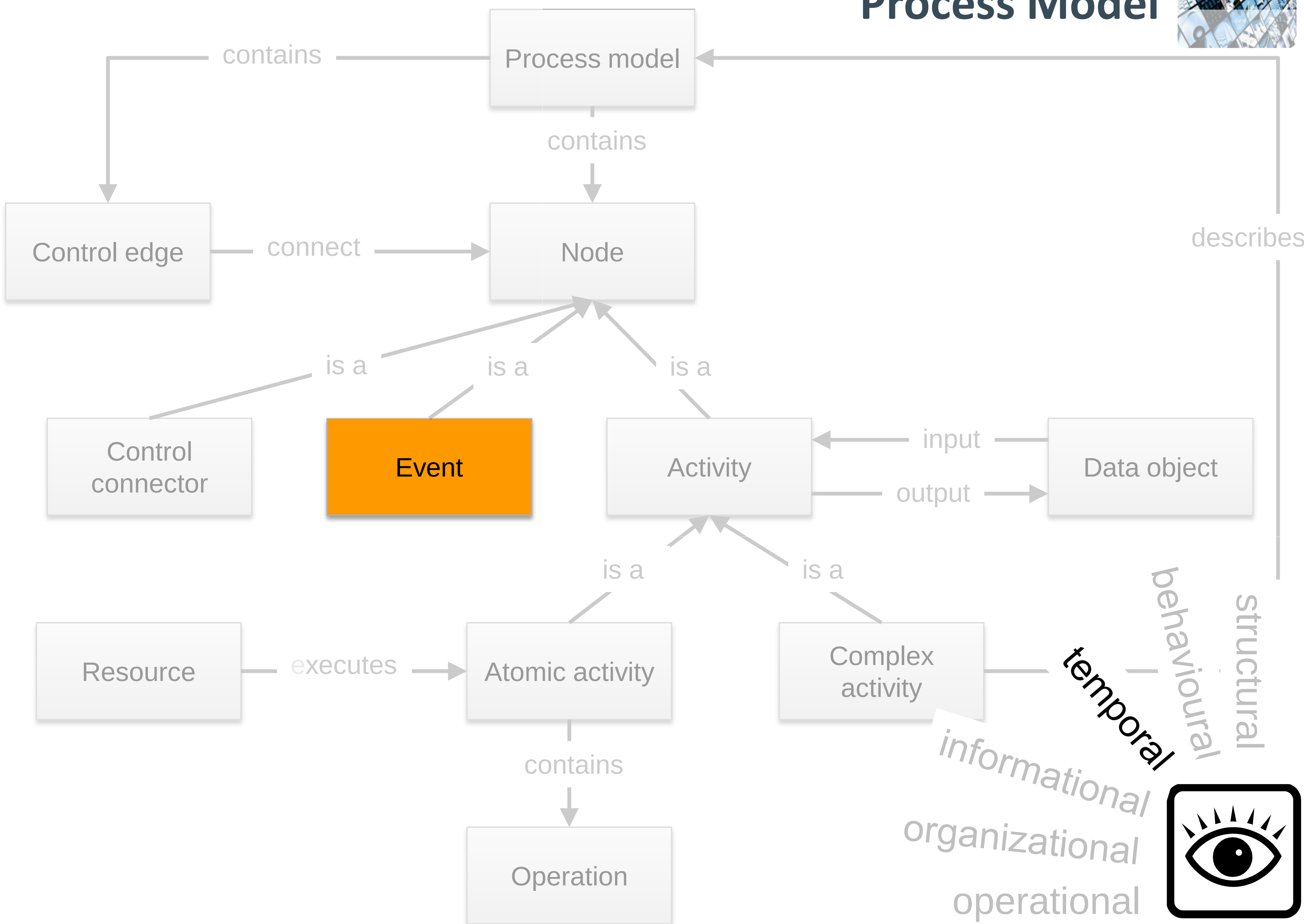
Process Model



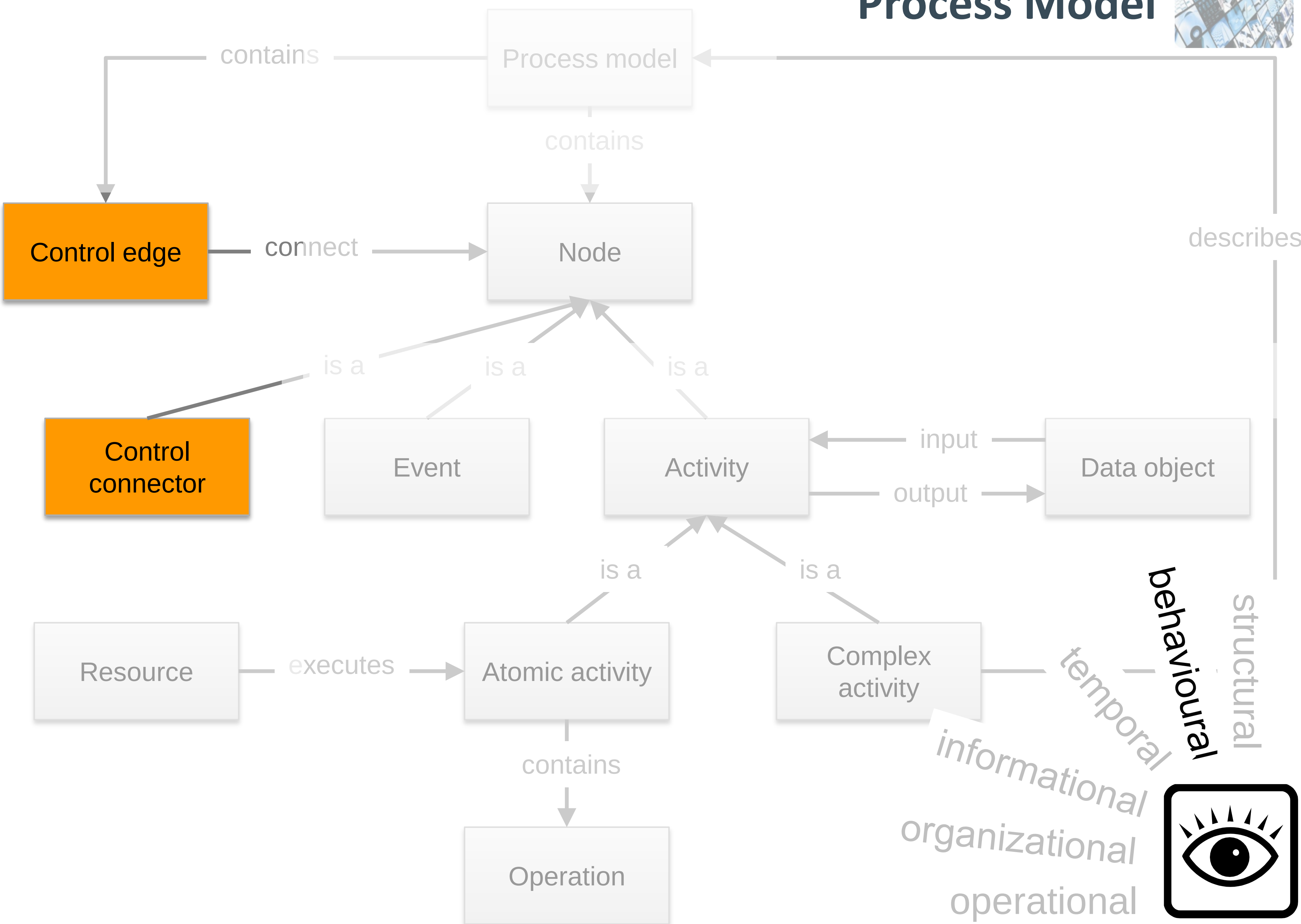
Process Model



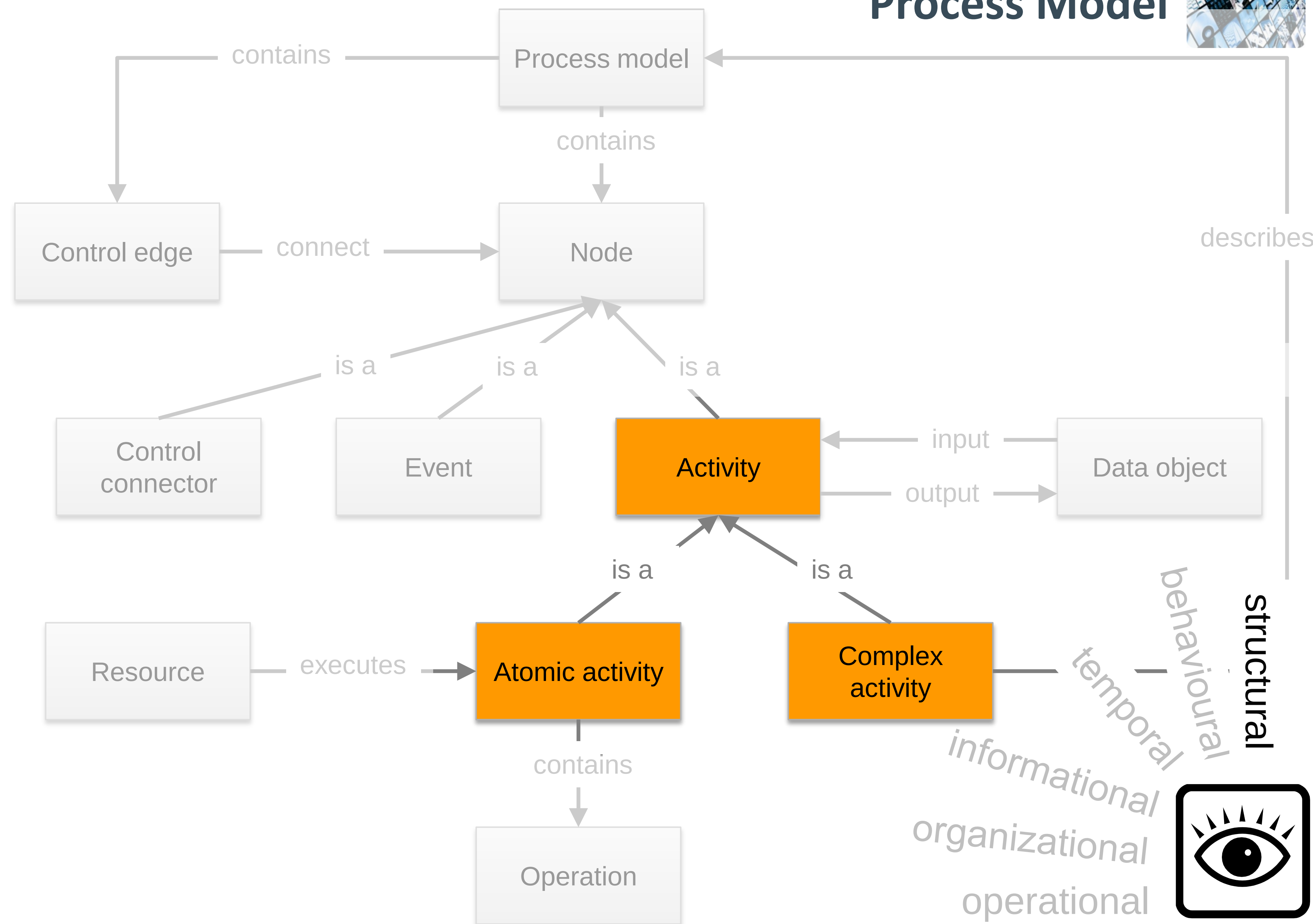
Process Model



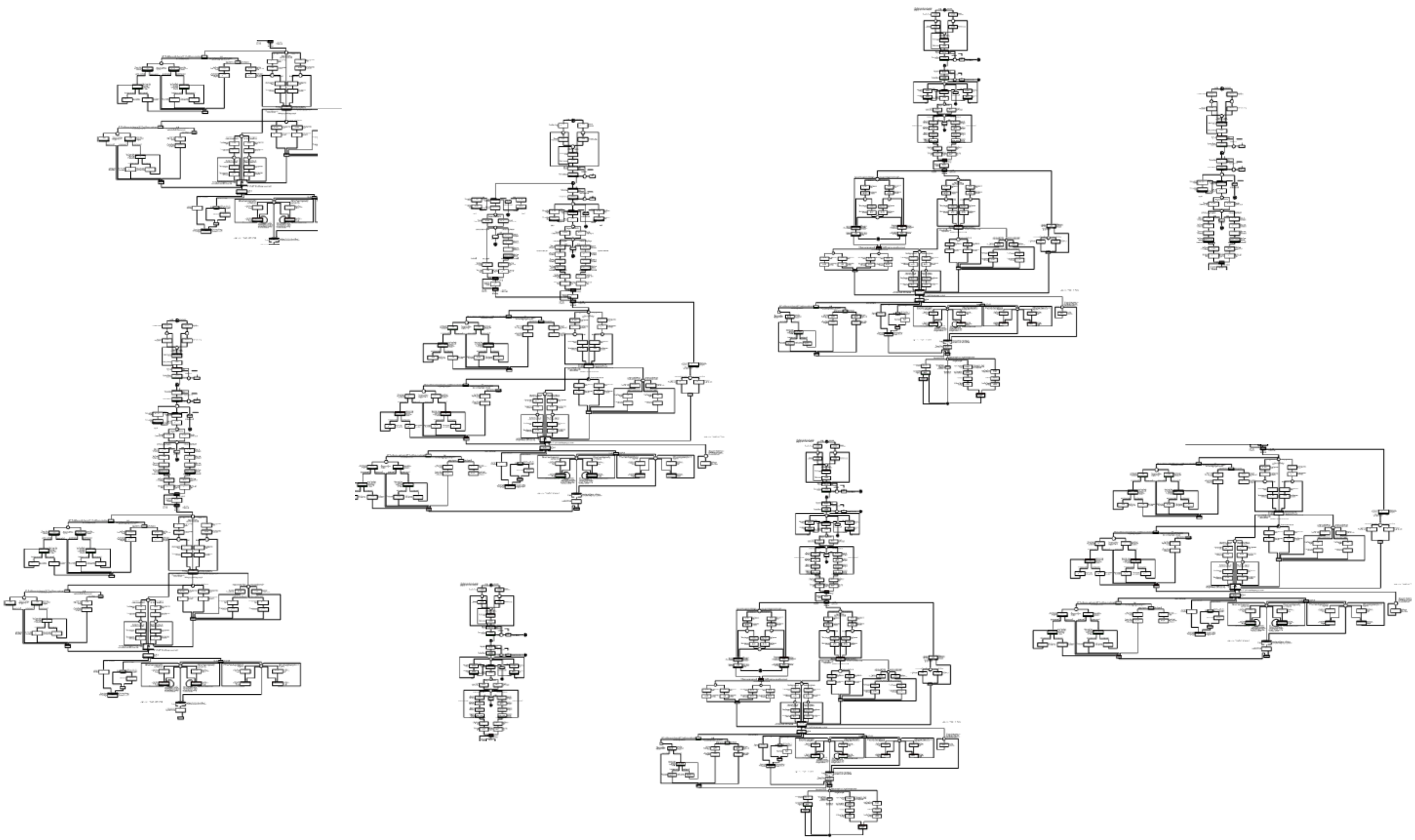
Process Model



Process Model



Process Model Collections

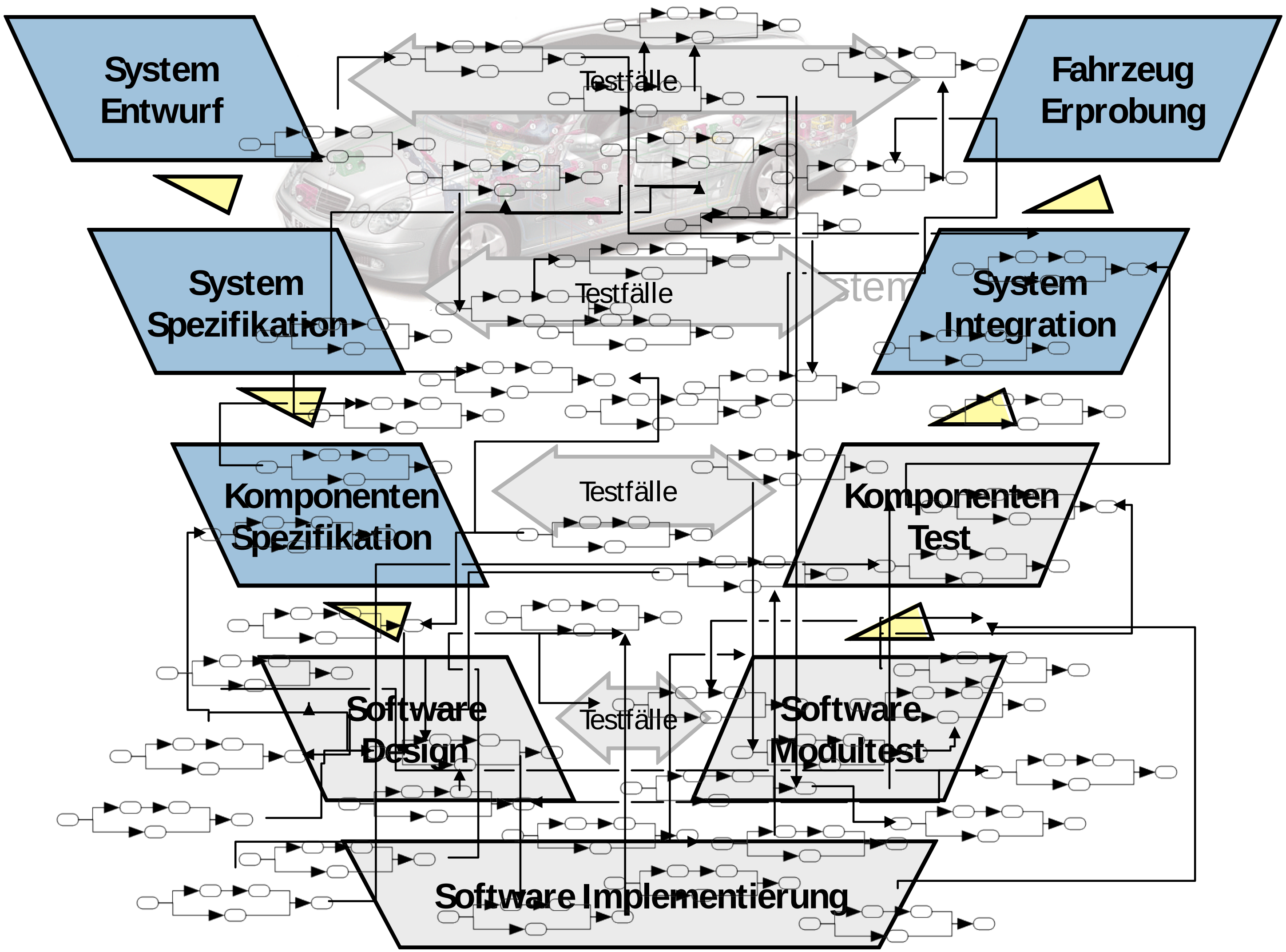


Process Model Collections

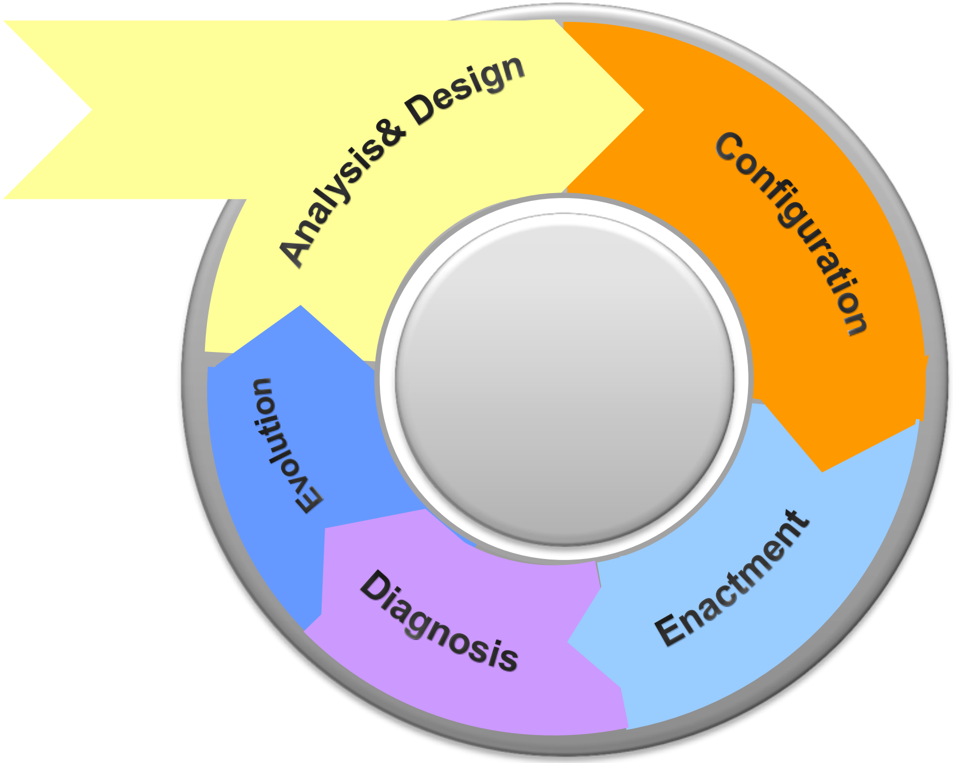
- Sets of process models
  - Sharing goals
    - Collections of process model variants
  - Targeted at different stakeholders
    - Collections of process model user views
  - Described at different abstraction levels
    - Collections of process model at different level of detail
  - Dealing with process model evolution
    - Collections of process model versions
  - Stored within the same repository
    - Collections of enterprise process models

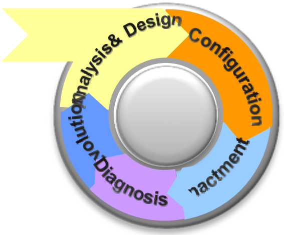
Process Structures

Process Structures



Lifecycle Phases



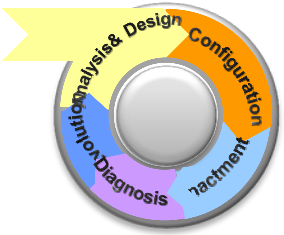


Lifecycle Phases

- Analysis & Design
  - BP identification and modelling
    - Based on domain requirements
    - BP Modelling notation and languages
  - Validation & Verification
    - Simulation techniques support Validation

» Resulting artefact: BP model

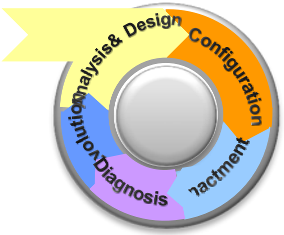
» Fostering communication between different stakeholders



Lifecycle Phases

- Configuration
  - Implementation of BP models
  - Implementation platform has to be chosen
  - BP model preparation for enactment
    - Interaction with the enterprise eco-system
      - users & existing systems
    - Tests to check desired behaviour

» Resulting artefact: Ready-to-enact BP model

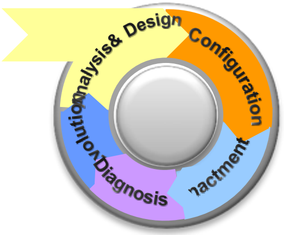


Lifecycle Phases

- Enactment
  - BP instance execution
    - Guaranteeing BP model constraints compliance
  - Monitoring & Visualizing techniques
    - Allow discovering the status of active BP cases

» Resulting artefacts:

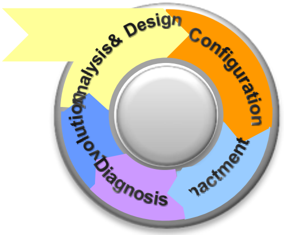
- » Business Process instances
- » Execution logs



Lifecycle Phases

- Diagnosis
  - Analysis of execution logs
    - Identification of poor quality designs
      - Fragments that are not used at all
    - Identification of problems regarding execution environment adaptation

» Resulting artefact: Process model and configuration changes report



Lifecycle Phases

- Evolution
  - – Application of changes to BP models based on
    - New requirements
    - Improvement opportunities

» Resulting artefact: BP Model more accurate to the BP and its environment

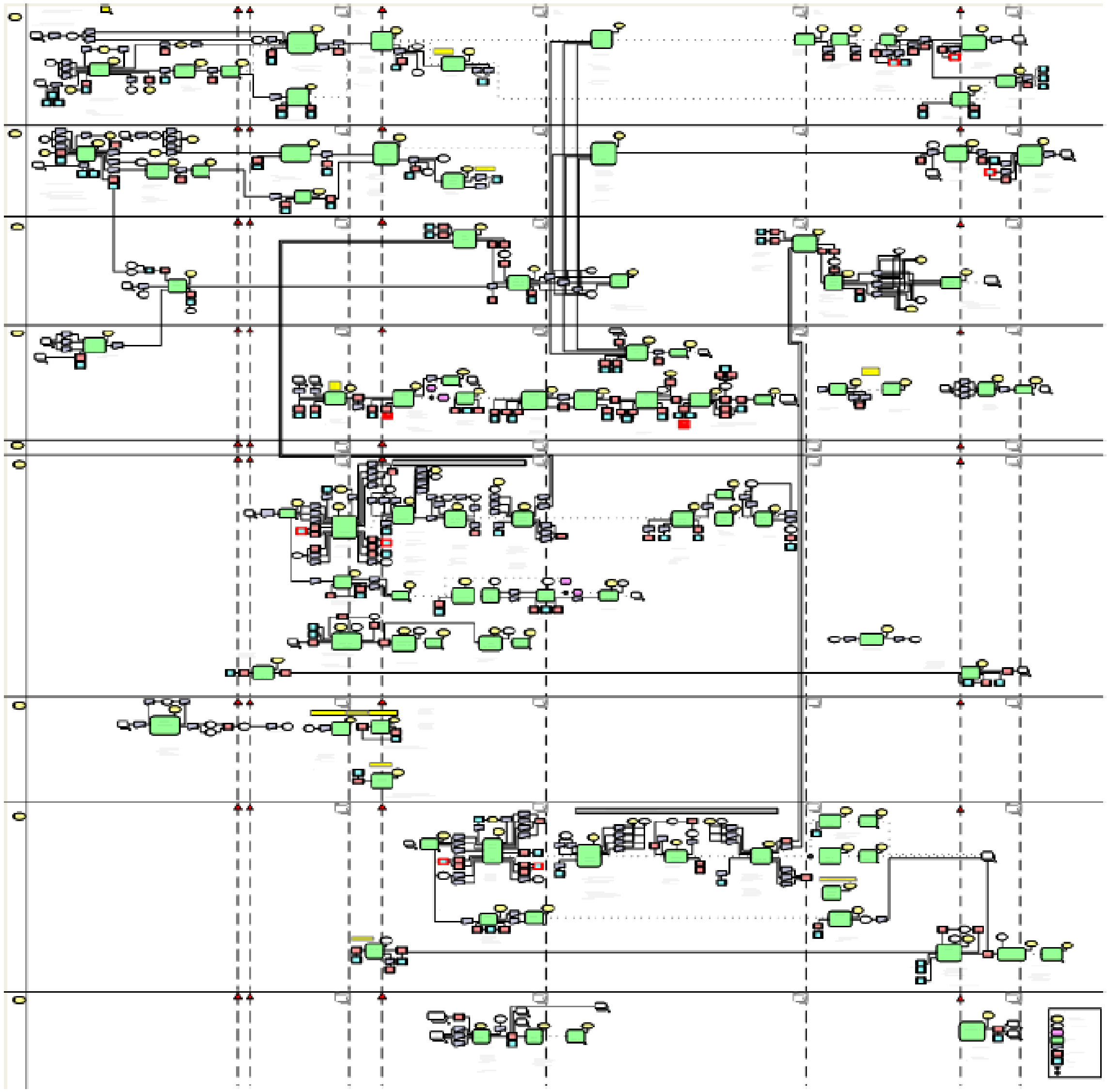
Agenda

- Challenges & Basic Notions
- Part I: Large Process Models
- Part II: Large Process Model Collections
- Part III: Large Process Structures
- References

Challenge: Large Process Models

- Challenge:

  Understanding process models of large size

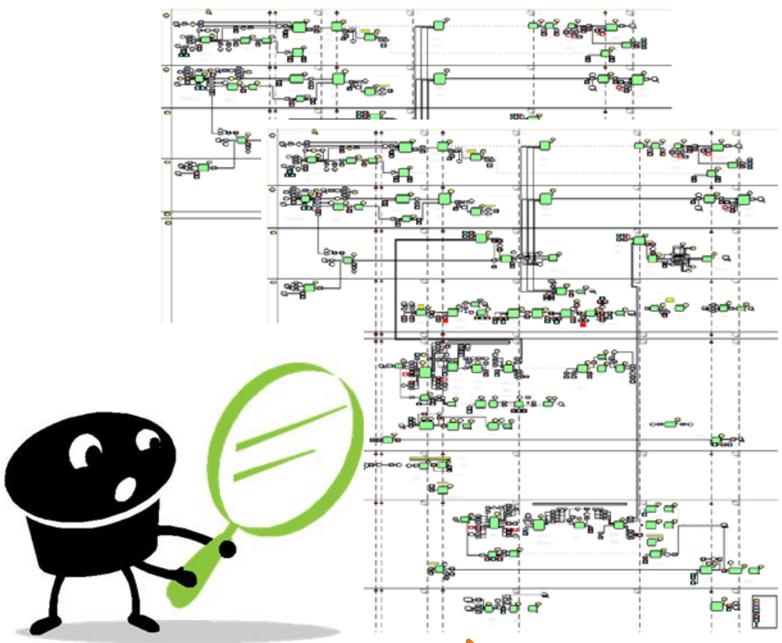


Process Model Smells

PMS1: Non-intention Revealing Naming of Activities / Process Model
PMS2: Contrived Complexity
PMS3: Redundant Process Fragment
PMS4: Large Process Models
PMS5: Lazy Process Models
PMS6: Unused Branches
PMS7: Frequently Occurring Instance Changes
PMS8: Frequently Occurring Variant Changes

(Weber and Reichert 2008, Weber et al. 2011)

Refactoring Large Process Models



Identification of Process Model Smells

Application of Refactoring Techniques

Process Model Refactorings

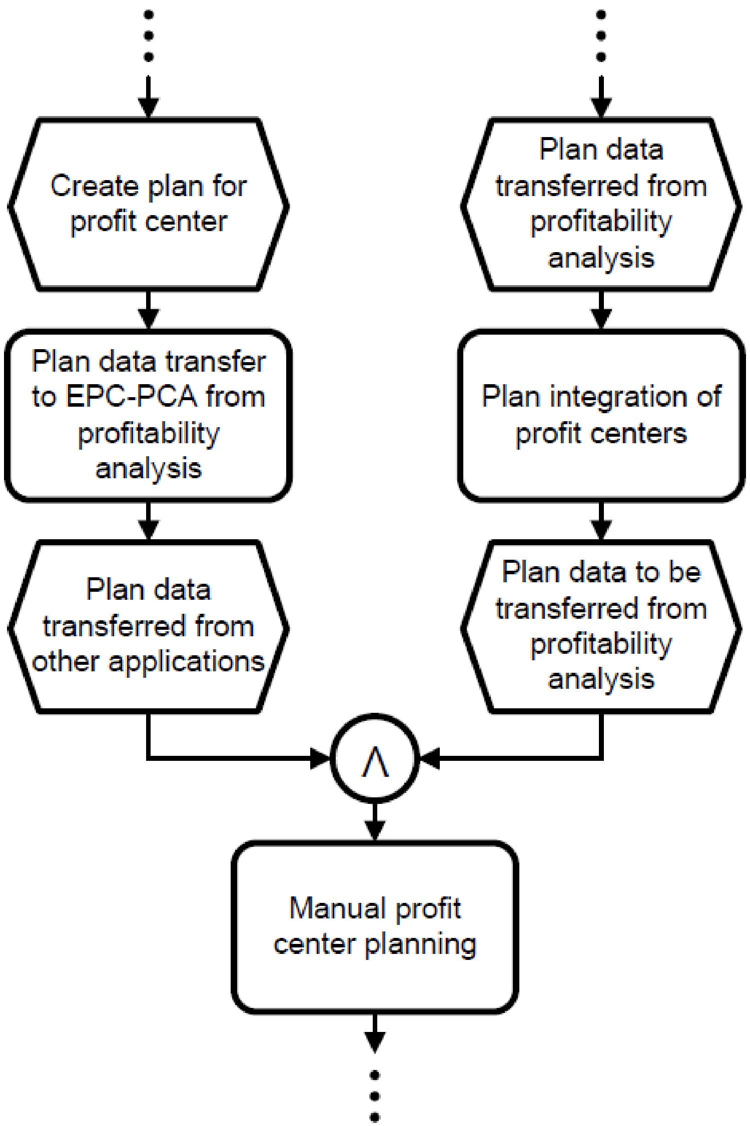
RF1: Rename Activity
RF2: Rename Process Schema
RF3: Substitute Process Fragment
RF4: Extract Process Fragment
RF5: Replace Process Fragment by Reference
RF6: Inline Process Fragment
RF7: Re-label Collection
RF8: Remove Redundancies

(Weber and Reichert 2008, Weber et al. 2011)

- PMS1. Non intention revealing naming of activities / process models
  - Ambiguous or non intention revealing labels
  - Inconsistent use of labeling styles

(Weber et al. 2011)

Example of Poor Labeling in SAP Reference Model



- Poor labeling may lead to ambiguities
  - Plan a data transfer?

    or

  - Transfer plan data?

Leopold et al. 2011

Number of ways students name an activity in a process model

- Insights from a process modeling experiment with 113 students.
- The following sentence in the process description

    "Afterwards the scouting team attends games of the player they are interested in live in the football stadium."

resulted into 84 different ways for naming this particular activity.

Fahland 2012

Guidelines for Labeling of Process Models

- Many guidelines stress the importance of proper labeling

  (Malone 2003, Sharp and McDermott 2008)

- Empirical insights show that verb-object style is best understandable

  (Mendling et al. 2010)

Automatic Refactoring of Labels

Labels are automatically refactored from action-noun style to labels in verb-object style

(Leopold, Smirnov, and Mendling 2012)

Plan data transfer to EPC-PCA from profitability analysis → Transfer plan data to EPC-PCA from profitability analysis

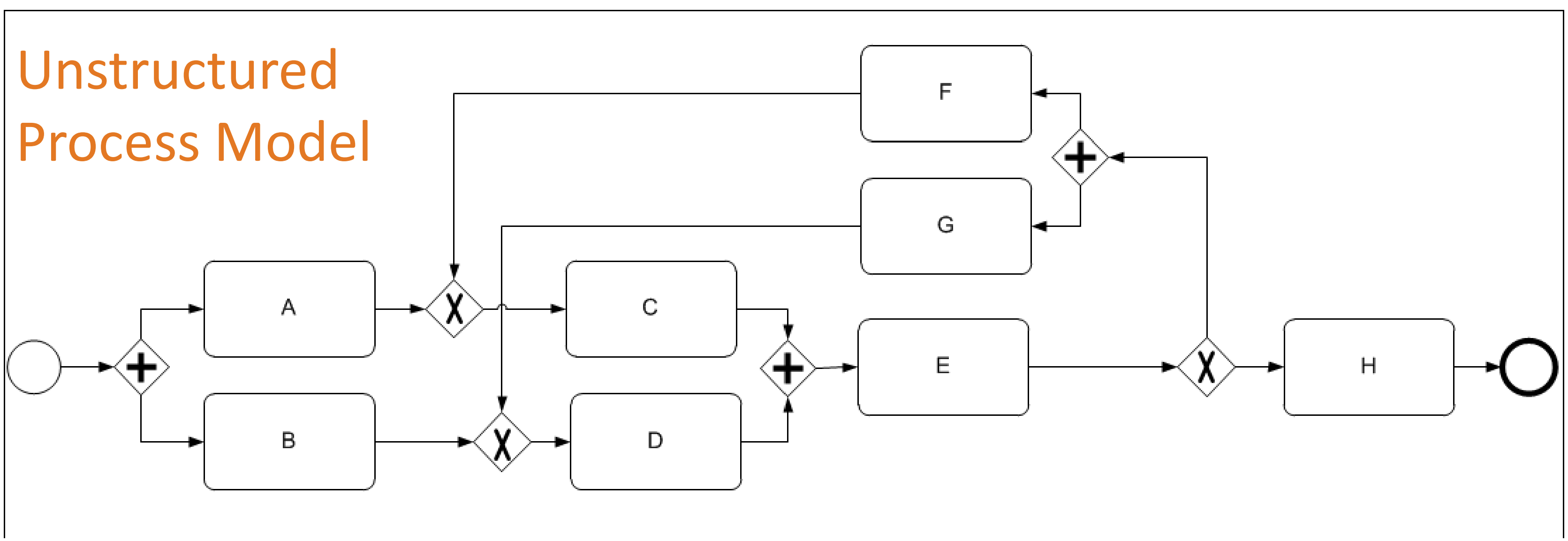
Refactorings RF1: Rename Activity

RF7: Re-label Collection

Structured versus Unstructured Business Process Models

- PMS2. Contrived Complexity
  - Unstructured process fragments are more difficult to understand than well-structured ones

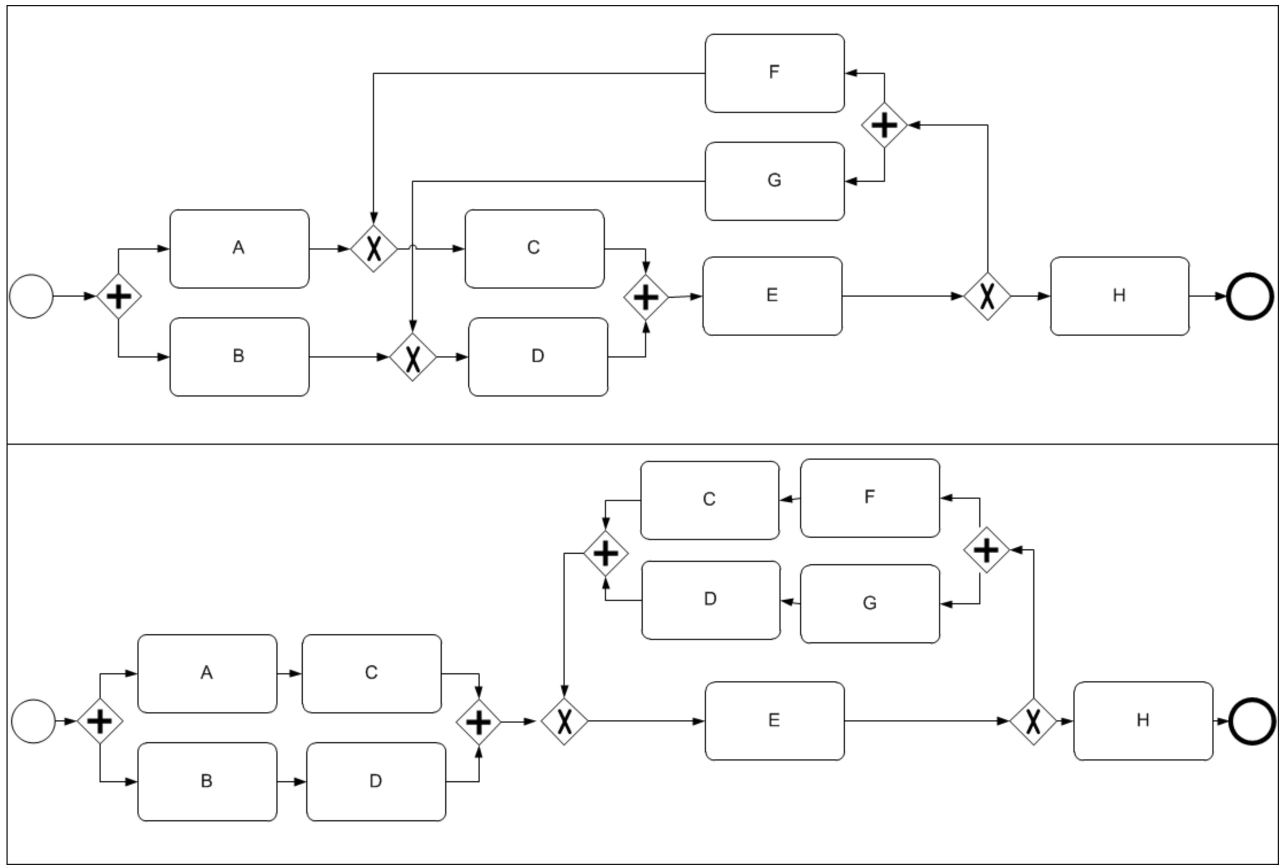
    (Mendling, Reijers, van der Aalst 2010)



- Automatic transformation of unstructured into structured model (Polyvyanyy et al. 2012)

Unstructured Process Model

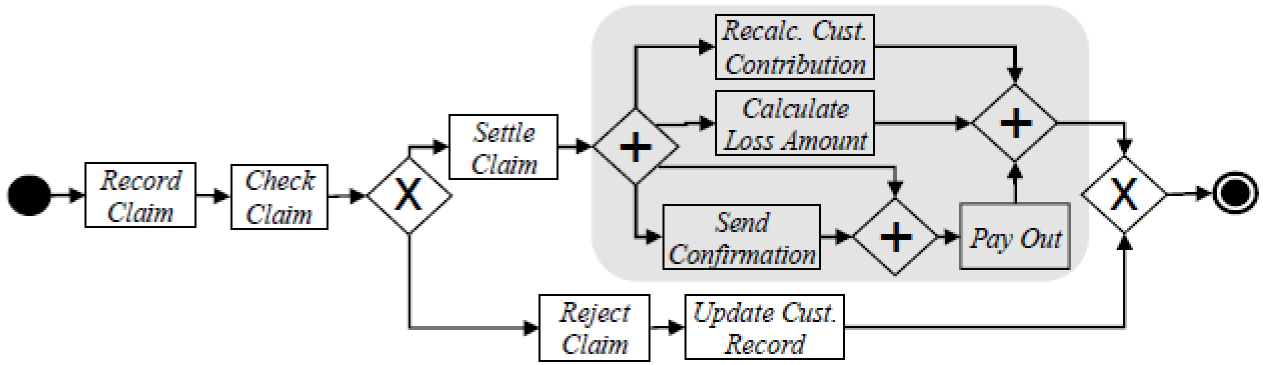
Well-structured Process Model

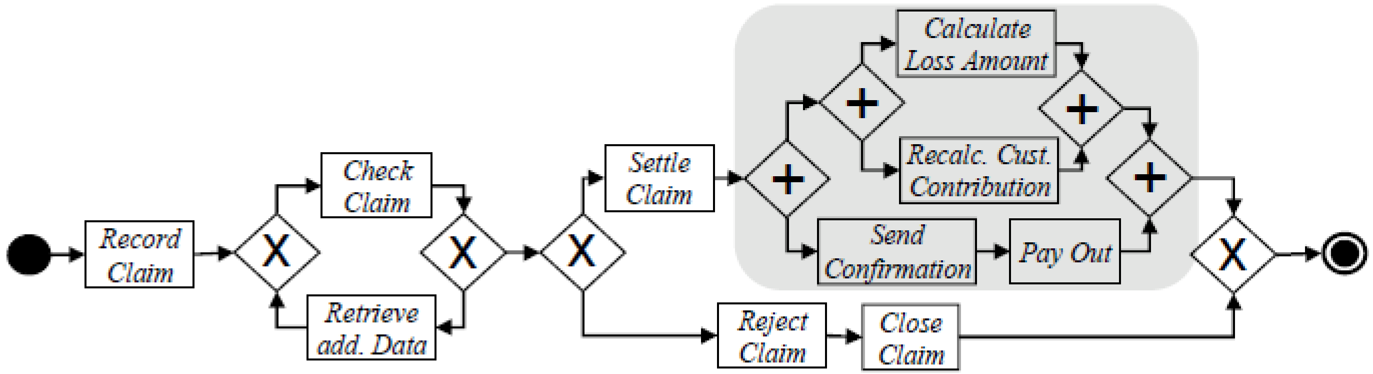


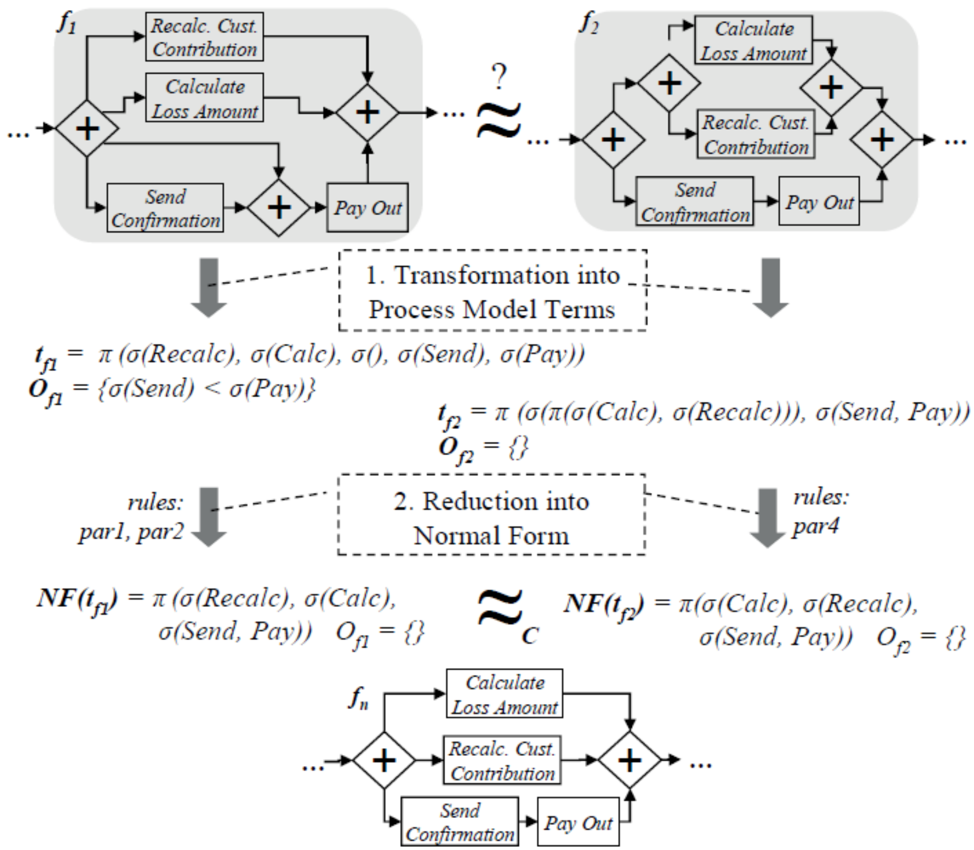
- Refactoring RF2: Substitute Process Fragment

Structured versus Unstructured Business Process Models

- PMS2. Contrived Complexity
  - Process Fragments are syntactically often different, but semantically equivalent (Gert et al. 2010)





- Automatic detection of semantically equivalent process fragments (Gert et al. 2010)
- 

- Refactoring RF2: Substitute Process Fragment

Redundant Process Fragments

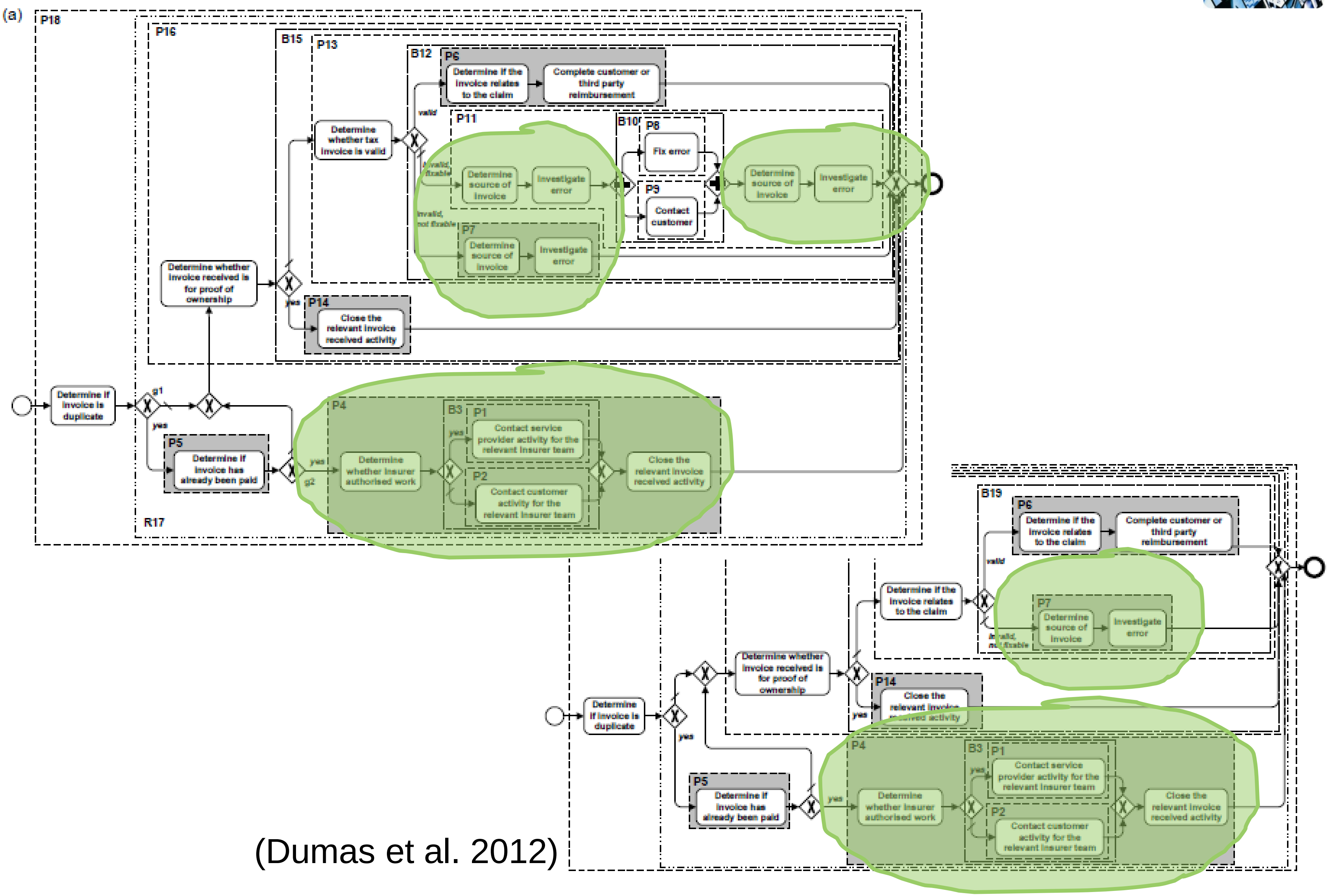
- PMS3: Redundant Process Fragments
  - Clones can be commonly found in existing process models

    (Dumas et al. 2012, Weber et al. 2011)

  - More than 560 clones in the SAP reference model

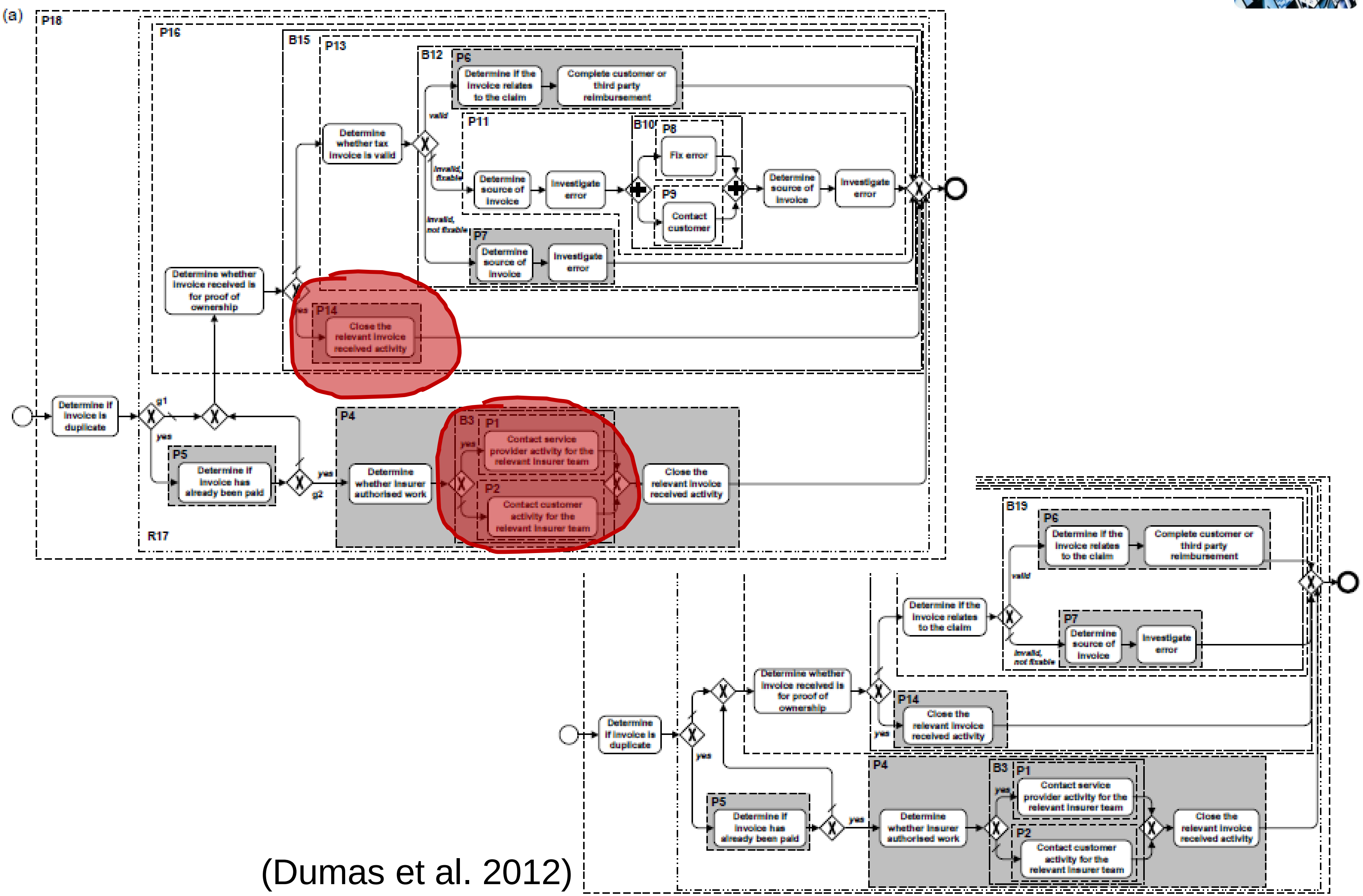
    (Dumas et al. 2012, Weber et al. 2011)

Redundant Process Fragments



(Dumas et al. 2012)

Redundant Process Fragments



(Dumas et al. 2012)

Redundant Process Fragments

- Method for automatically detecting exact clones

  (Dumas et al. 2012)

- Detected clones can be automatically extracted to sub processes

- Refactoring RF4: Extract Process Fragment

  RF5: Replace Process Fragment by Reference

  RF8: Remove Redundancies

Large Process Models

- PMS4: Large Process Models
  - Literature reports about process models with several hundred activities (Soto et al. 2008)
  - Large process models tend to comprise more formal flaws than smaller ones (Mendling et al. 2008)

Large Process Models

- Method for automatic modularization of business process models (Reijers et al. 2011)

- Method for the automatic labeling of process models (Leopold et al. 2011)

- Refactoring RF4: Extract Process Fragment
  RF5: Replace Process Fragment by Reference
  RF8: Remove Redundancies

Hierarchy in Conceptual Models
Common Belief versus Empirical Evidence

- Modularization is a widely used strategy to reduce complexity in conceptual models
- Hierarchical structures supported by many conceptual modeling languages
- Empirical evidence on the benefits of hierarchy are contradictory

Hierarchisation in Conceptual Models
A Systematic Literature Review

Work	Findings
Moody [15] Domain: ER-Models	Positive influence on accuracy, no influence / negative influence on time
Reijers et al. [16, 17] Domain: Business Process Models	Positive influence on understandability for one out of two models
Cruz-Lemus et al. [9, 18] Domain: UML Statecharts	Series of experiments, positive influence on understandability in last experiment
Cruz-Lemus et al. [13] Domain: UML Statecharts	Hierarchy depth of statecharts has no influence
Shoval et al. [14] Domain: ER-Models	Hierarchy has no influence
Cruz-Lemus et al. [8] Domain: UML Statecharts	Positive influence on understandability for first experiment, negative influence in replication
Cruz-Lemus et al. [12, 19] Domain: UML Statecharts	Hierarchy depth has a negative influence

(Zugal et al. 2011a)

Hierarchisation in Conceptual Models
A Systematic Literature Review

Work	Findings
Moody [15] Domain: ER-Models	Positive influence on accuracy, no influence / negative influence on time
Reijers et al. [16, 17] Dom	Positive influence on understandability for one
Cru Do	
Cru Do	
Sh Do	
Cruz Domain: UML Statecharts	experiment, negative influence in replication
Cruz-Lemus et al. [12, 19] Domain: UML Statecharts	Hierarchy depth has a negative influence

Under which circumstances can positive / negative influcences on understandability be expected?

- Answering questions about a model is a *problem solving task*

- Three different problem solving processes (Larkin and Simon 1987)
  - Search
  - Recognition
  - Inference

Limitations of Working Memory

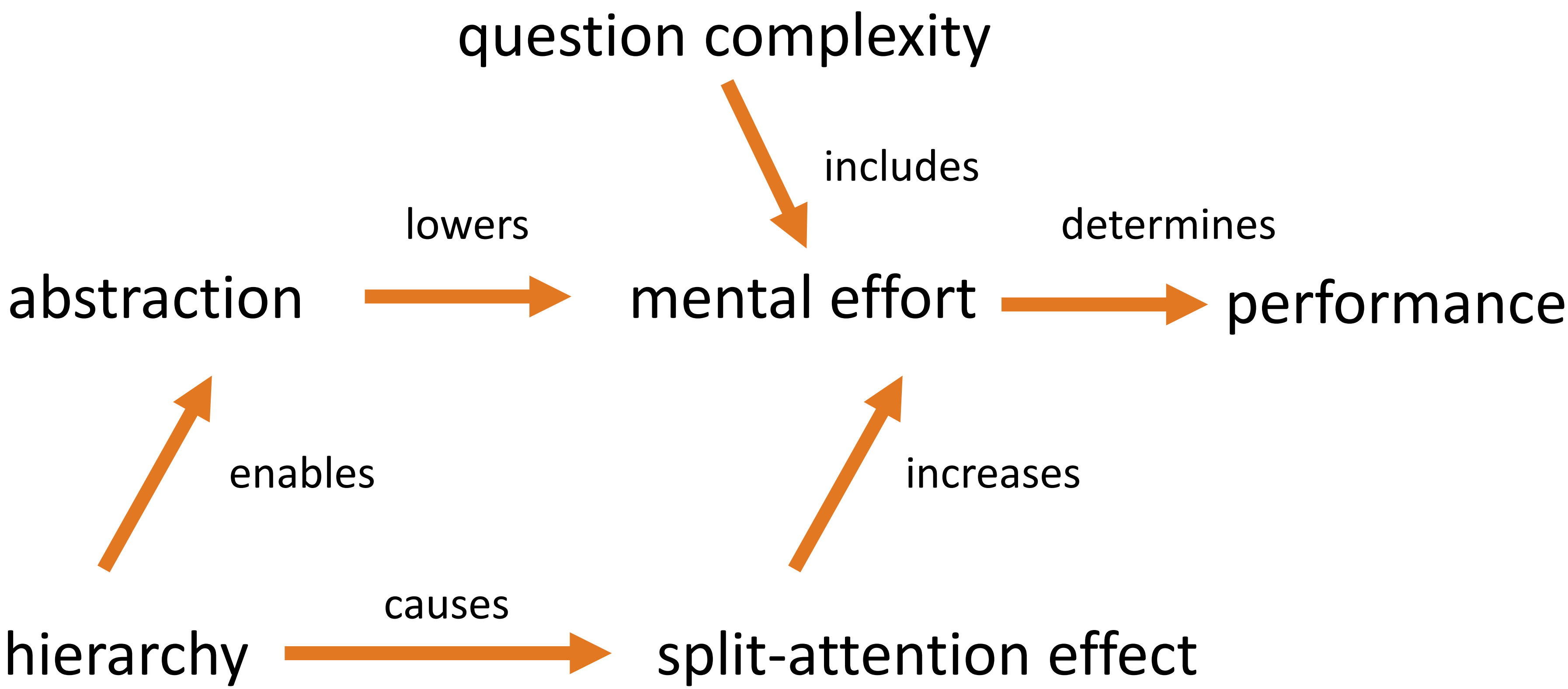
- Central Concept: Working Memory
  - Required by all conscious mental activities
  - Mental effort: utilization of working memory
  - Severely limited: 7 +/- 2 information „slots“ (Miller 1956)
  - Overflow: rapid performance decrease! (Sweller 1988)
  - Mental effort: amount of working memory used (Paas et al. 2003)

Two Opposing Forces

- **Abstraction**
  - hiding of irrelevant information (Parnas 1972)
  - supports human mind's attention management (Larking and Simon 1987)

- **Split-attention effect (Sweller and Chandler 1994)**
  - Occurs when information from several sources needs to be integrated
  - switching attention between models

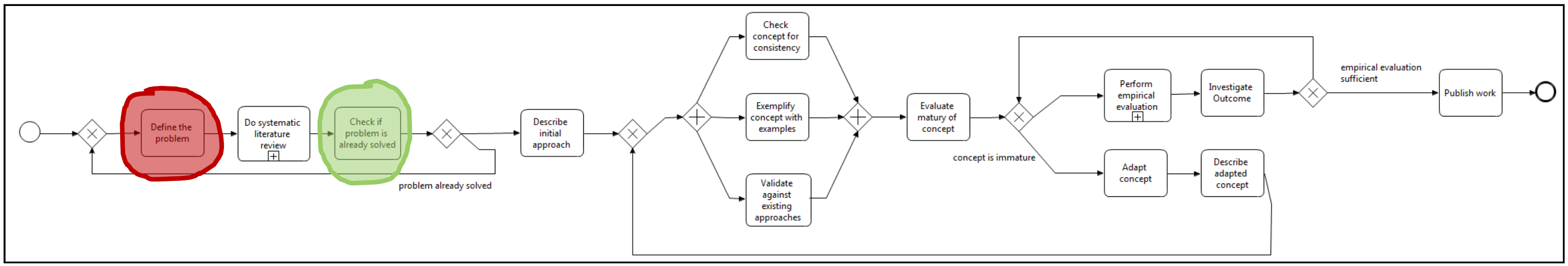
Theoretical Framework for Measuring Understandability



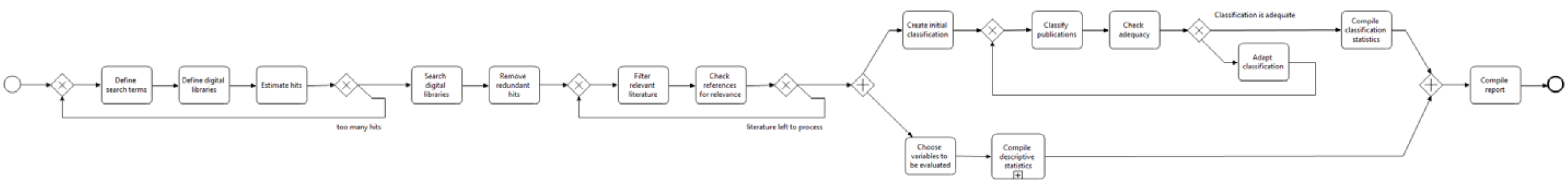
(Zugal et al. 2011a)

Hierarchical Model – Abstraction Question

- 'Define the problem' can be executed after 'Check if problem is already solved'.

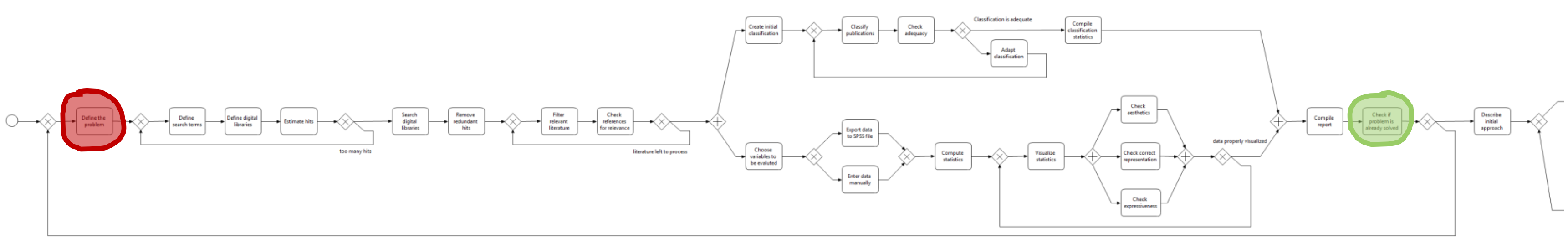


Sub Process: Do Systematic Literature Review



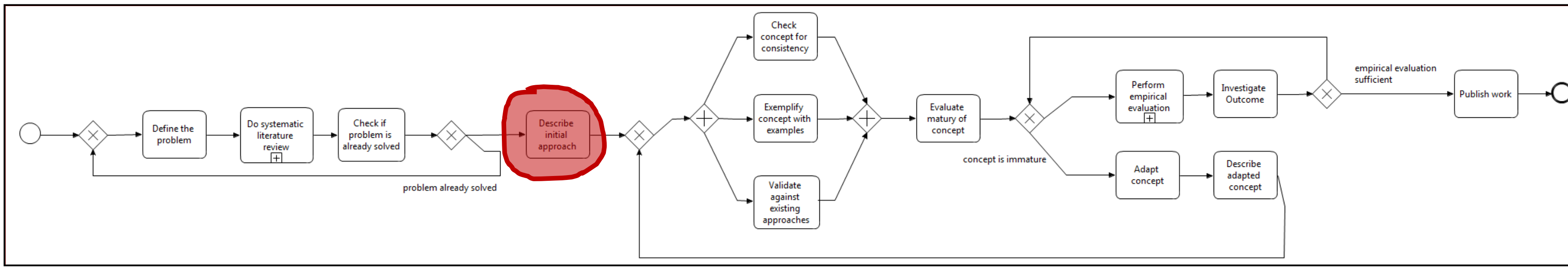
Flat Model – Abstraction Question

- 'Define the problem' can be executed after 'Check if problem is already solved'.

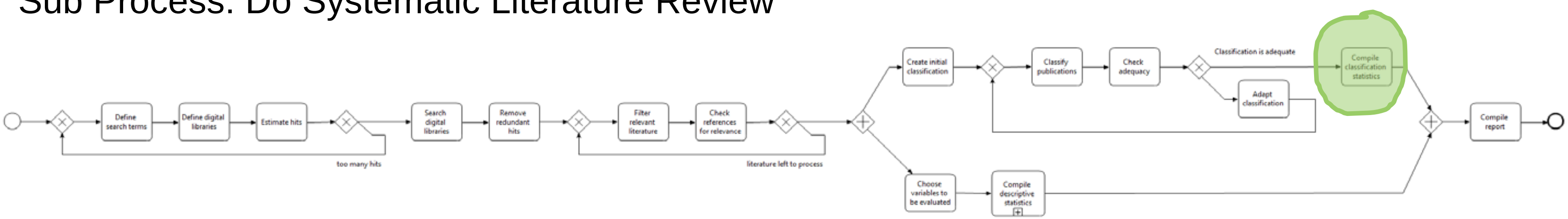


Hierarchical Model – Fragmentation Question

- 'Describe initial approach' can be executed after 'Compile classification statistics'.

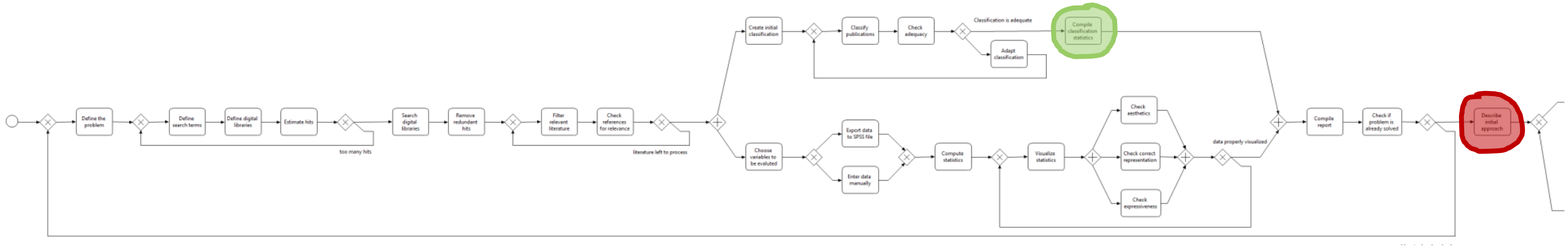


Sub Process: Do Systematic Literature Review



Flat Model – Fragmentation Question

- 'Describe initial approach' can be executed after 'Compile classification statistics'.



Hierarchisation is Always a Trade-off

- Modularization is always a trade-off!

- Also the optimal modularization can only increase the “average” understanding!
  - Some questions become easier to answer
  - Some questions become harder to answer

(Zugal et al. 2011a, Zugal 2011b, Zugal 2011c)

Hierarchisation is Always a Trade-off

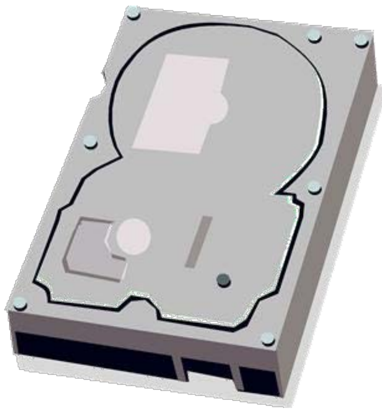
- Modularization is always a trade-off!
- Also the optimal modularization can only increase

Dynamic modularization / visualization / navigation is needed !

(Zugal et al. 2011a, Zugal 2011b, Zugal 2011c)

The Role of External Memory

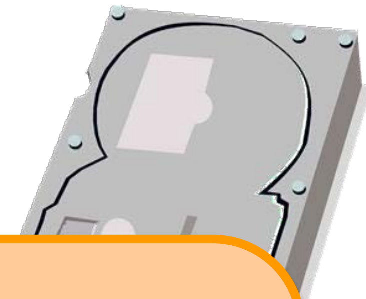
- External memory
  - mechanism for reducing mental effort
  - Information storage outside the human cognitive system (e.g., pencil and paper or a blackboard)



- Cognitive Trace
  - Information taken from working memory and stored in an external memory (e.g., to mark, update, and highlight information)

The Role of External Memory

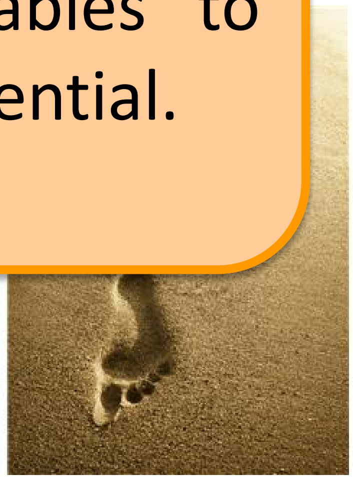
- External memory
  - mechanism for reducing mental effort
  - Information storage outside the human

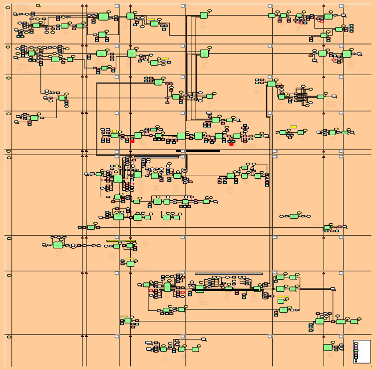


For process models of large size tool support, which enables to offload information, is essential.

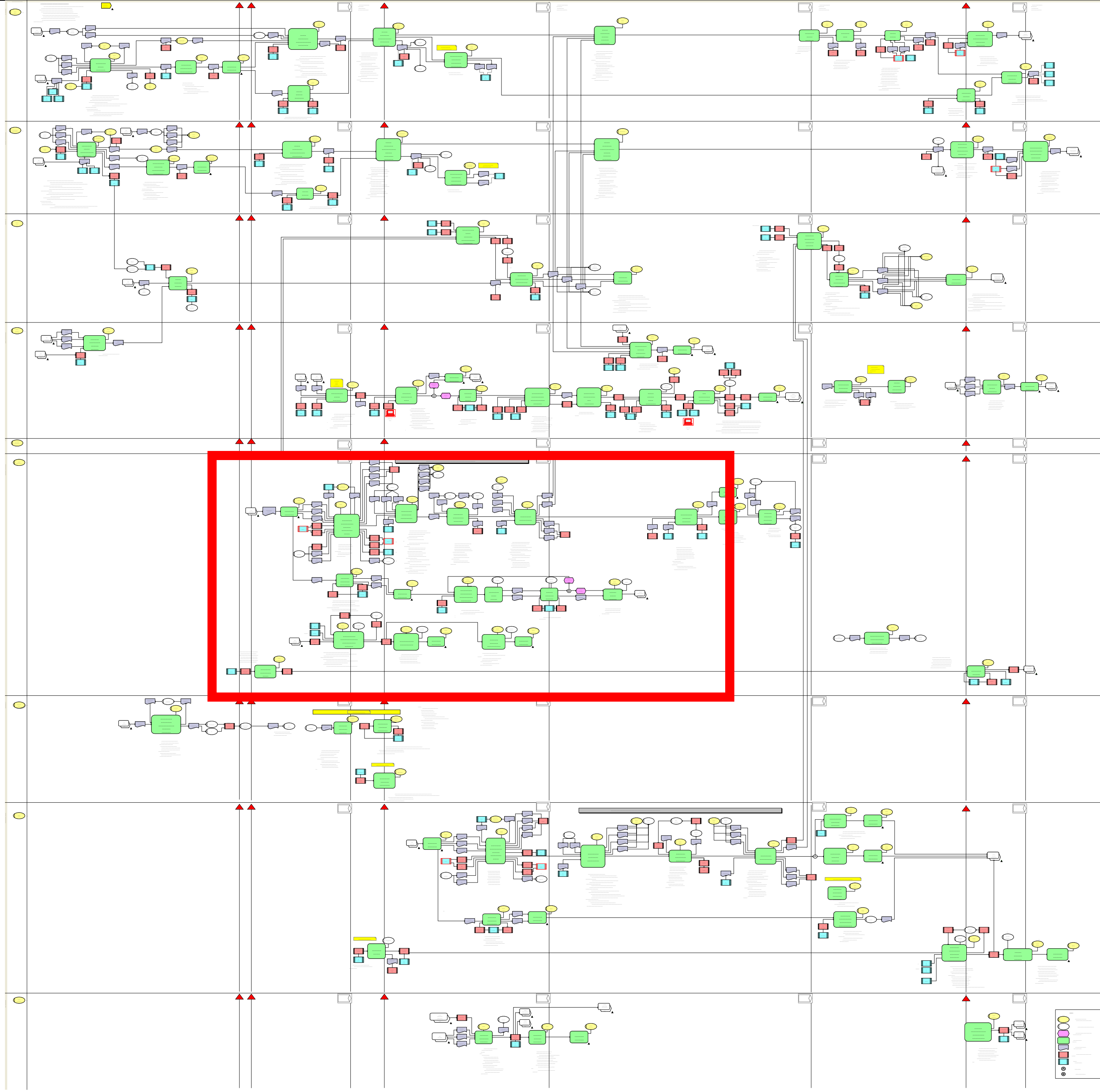
-

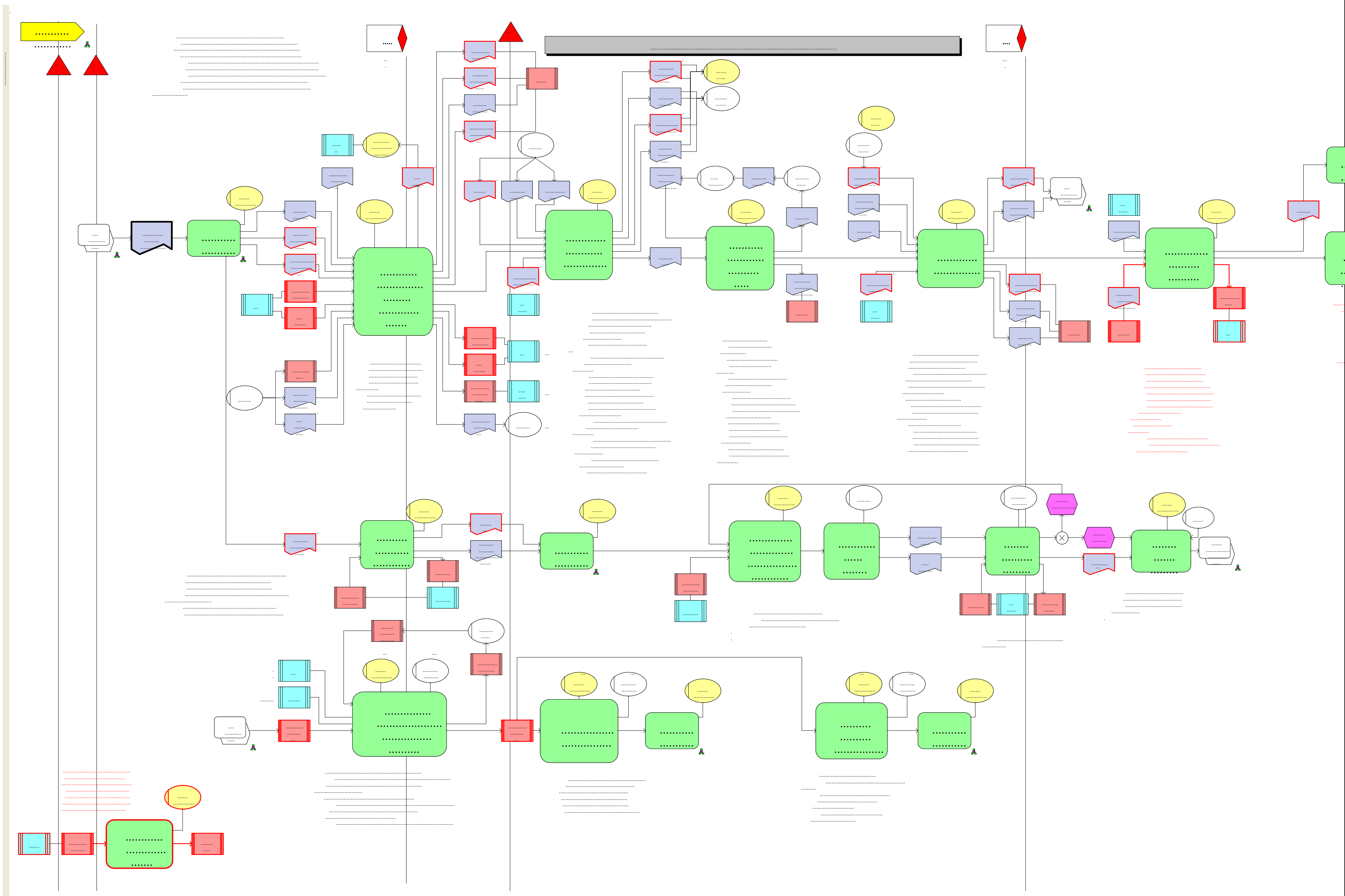
mark, update, and highlight information)



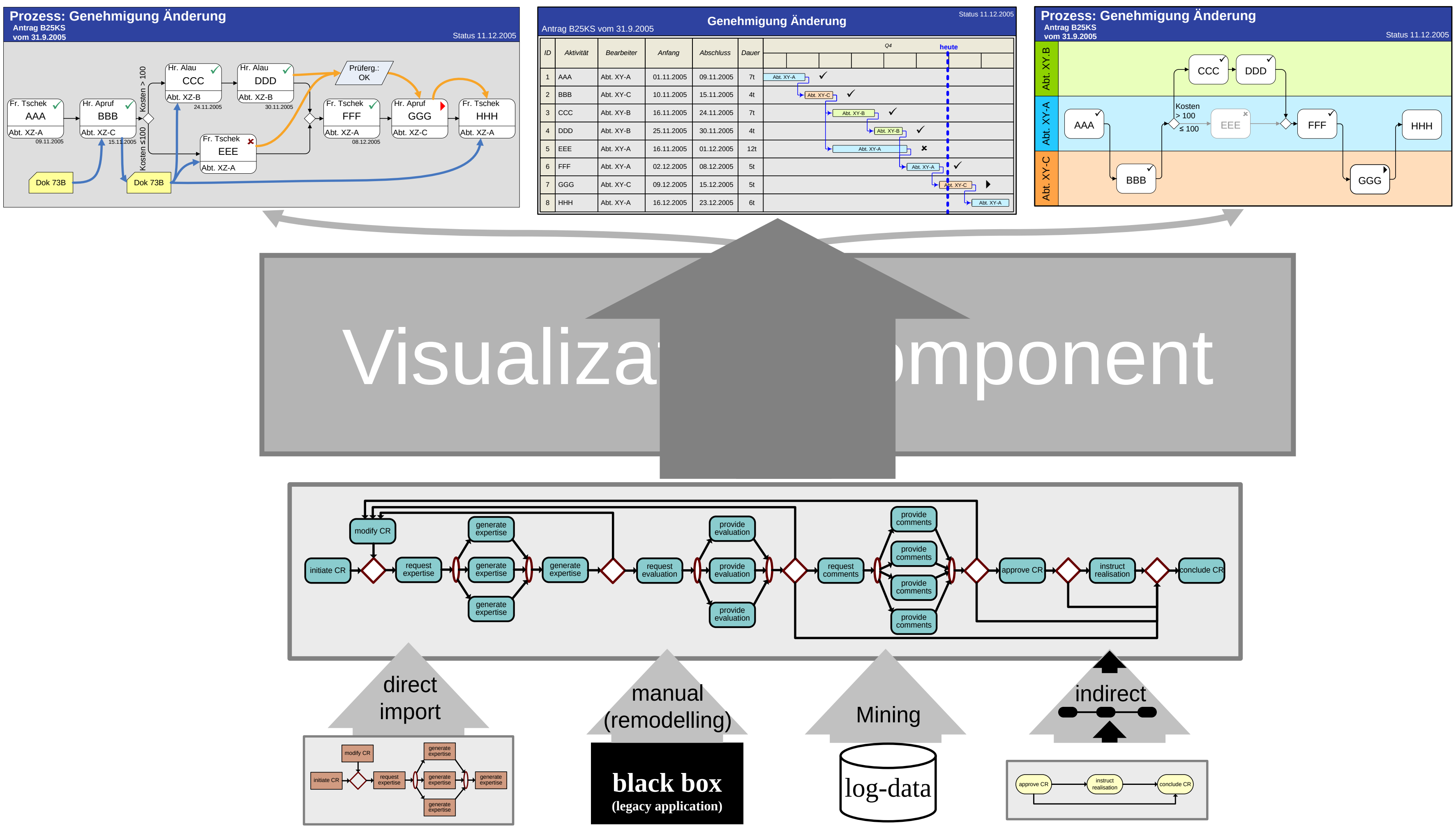


Visualizing and Abstracting Large Process Models

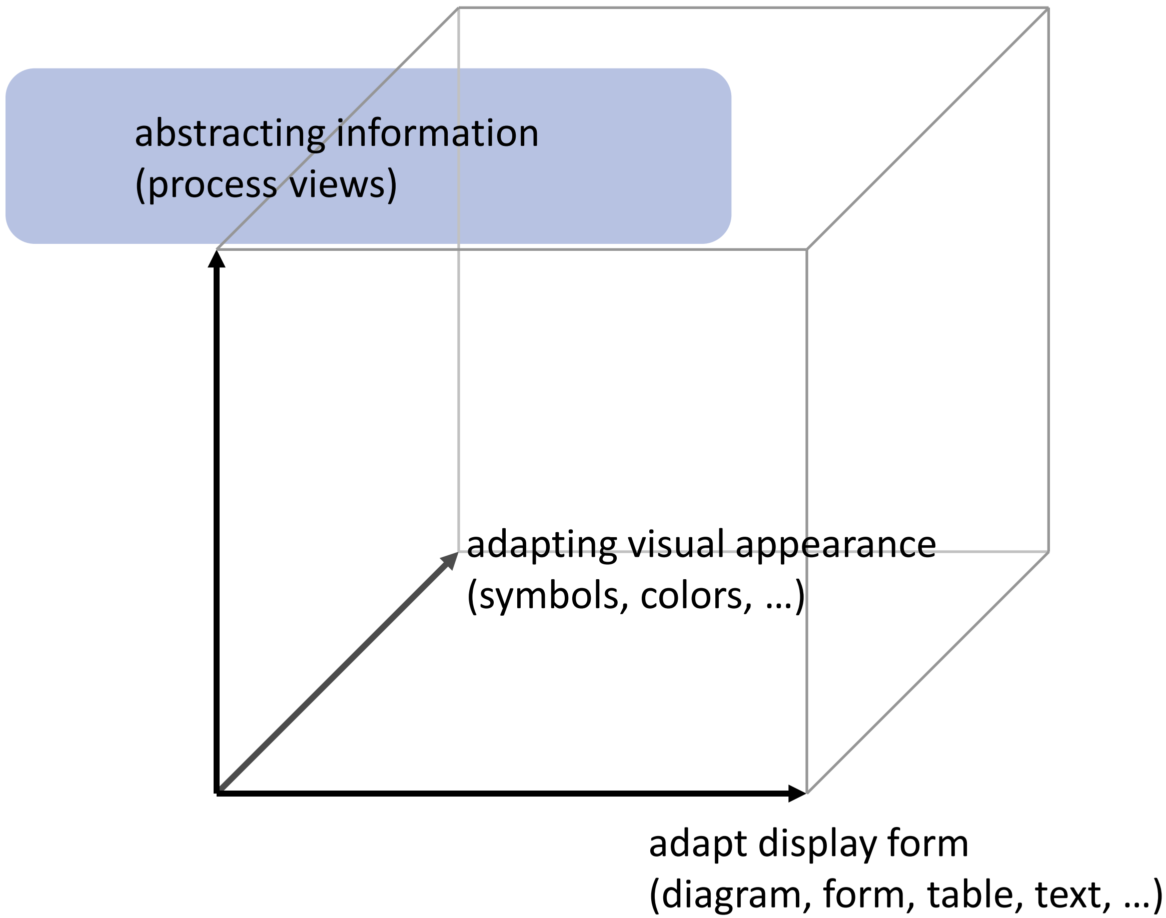




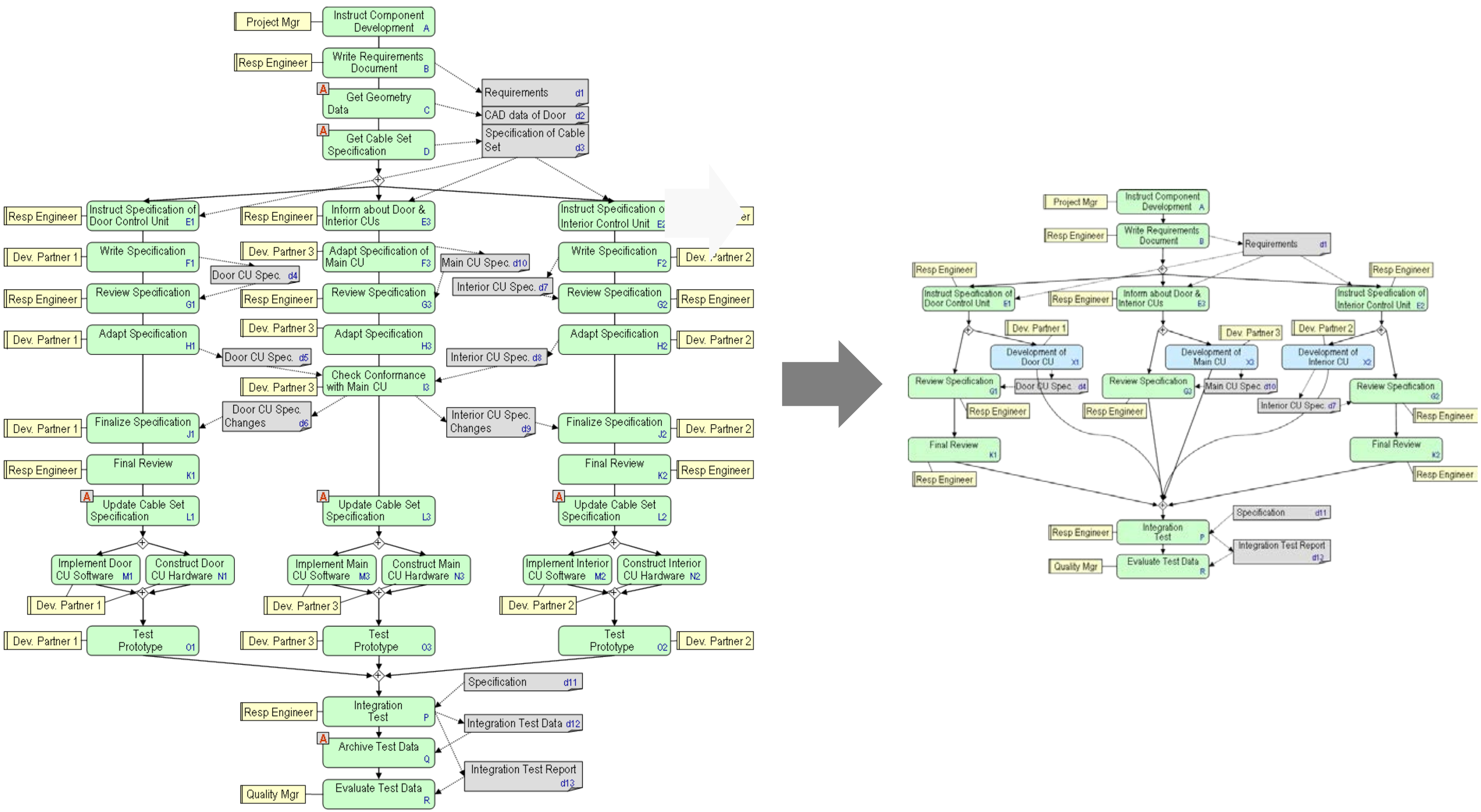
Process Visualization
What is needed?



Process Visualization Dimensions



Process Visualization
Abstracting Process Models: Goals



Abstracting Process Models: Goals

Goals:

- Decreasing the complexity of (large) process models
- Eliminating or abstracting process information

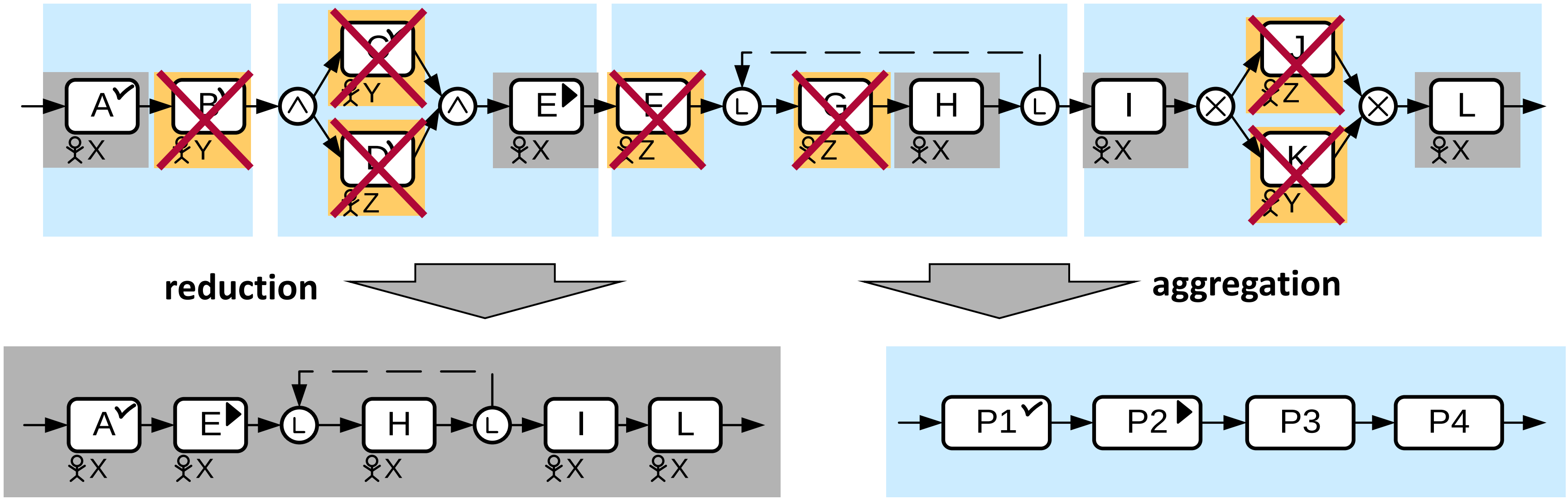
▶ **Personalize process models through process views**

- Process views should be …
  - easy to define
  - dynamically built if required

„Only show my activities!“

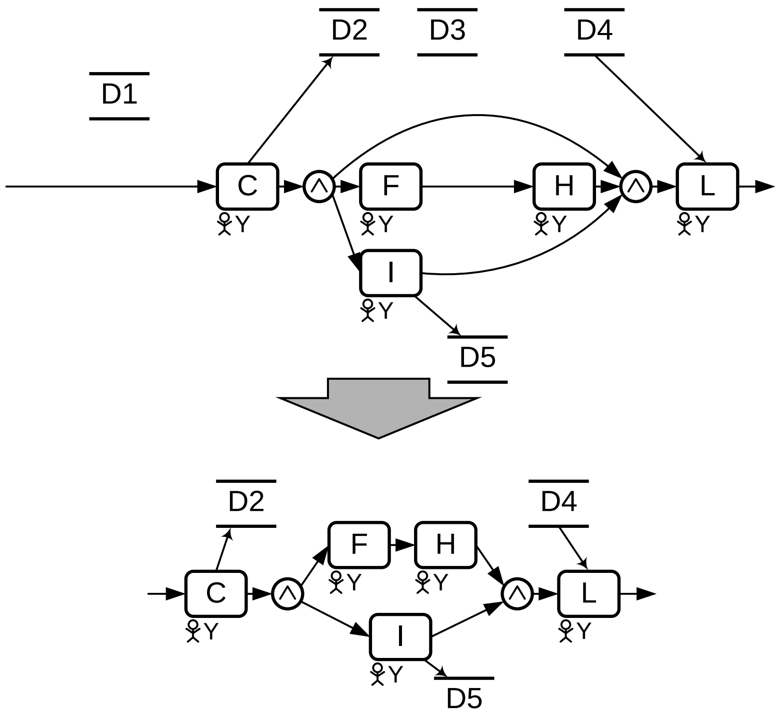
„Do not display technical activities“

„Aggregate completed parts“



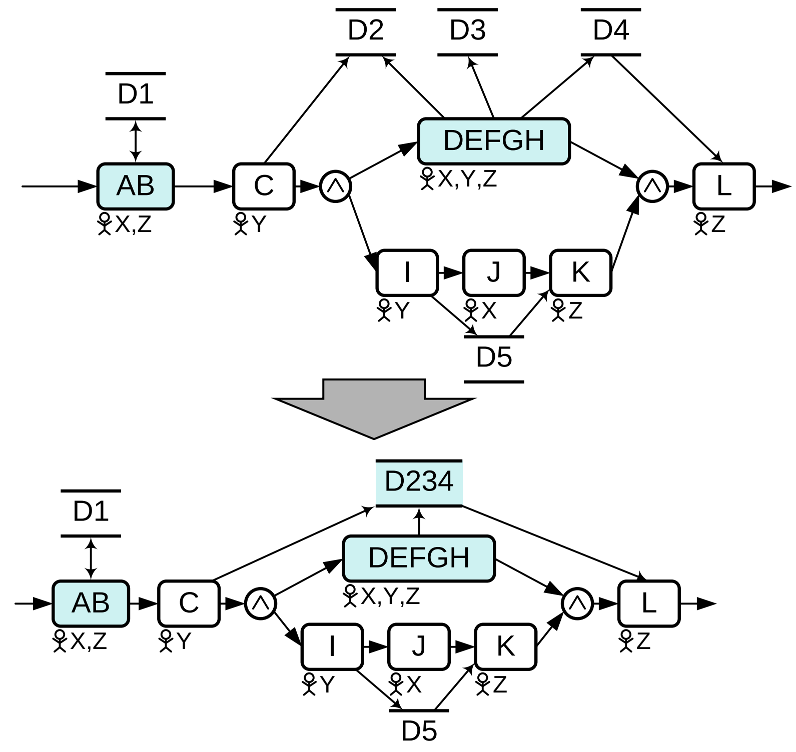
Process Visualization
Abstracting Process Models: Fundamental Techniques

Reduction



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

Aggregation



- Aggregate activities
- Aggregate adjacent objects if required

Bobrik et al., 2007

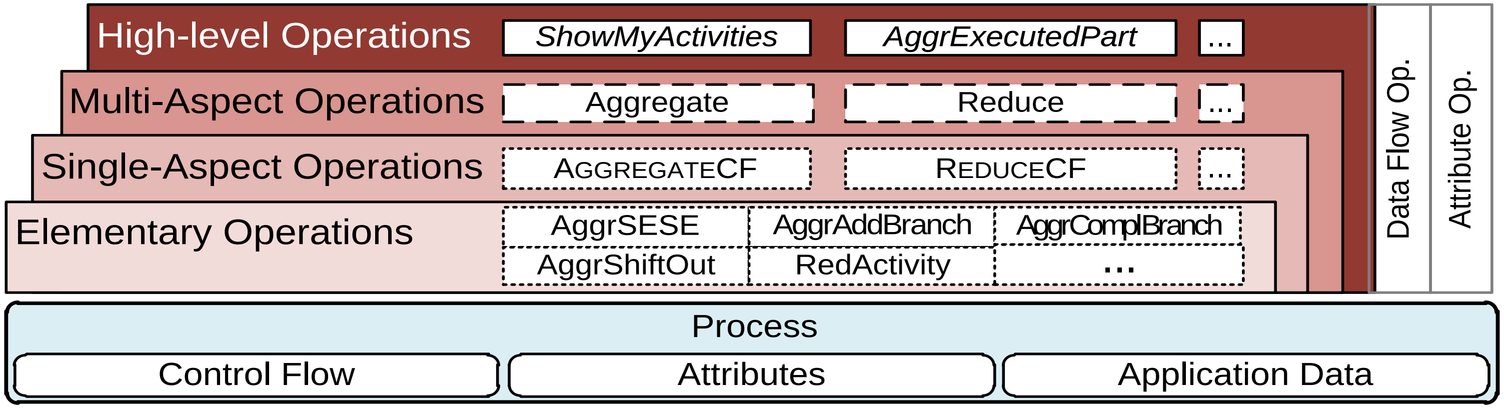
Process Model Abstraction and Process Views in Proviado

(Bobrik, Bauer, & Reichert, 2007; Reichert et al., 2012)

- Related approaches, e.g.,
  - Smirnov, Reijers & Weske, 2011
  - Eshuis & Grefen, 2008
  - Schumm, Latuske & Leymann, 2011

Abstracting Process Models: The Proviado Approach

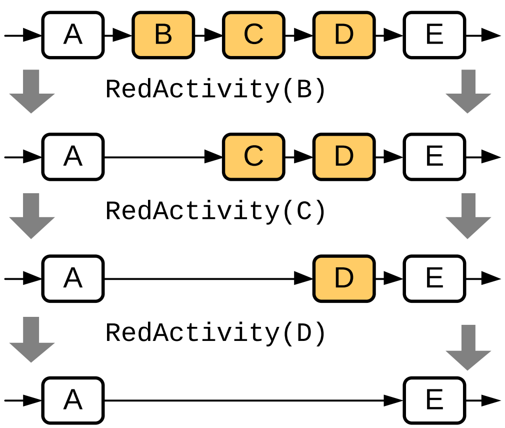
- A multi-layer approach for abstracting process models and building process views



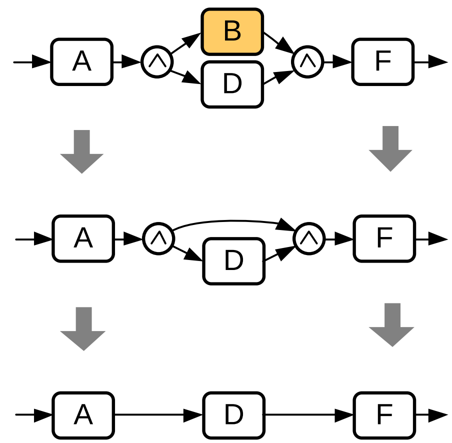
Process Visualization
Abstracting Process Models: Elementary Operations

Elementary reduction operations

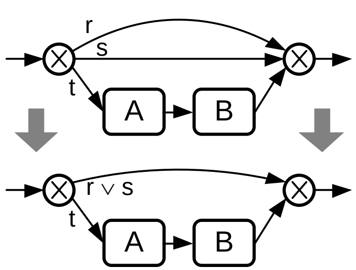
ReduceCF({B,C,D})

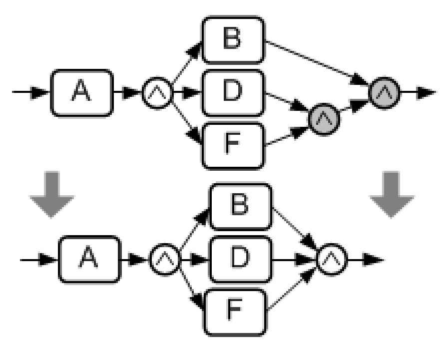


RedActivity(B)

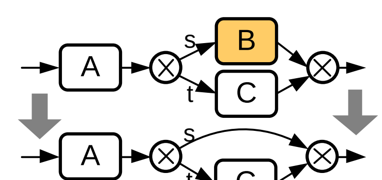


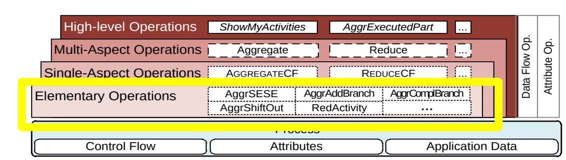
Further refactorings:





but …

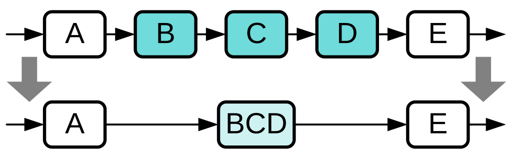




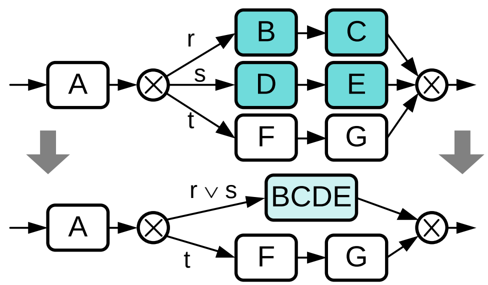
Process Visualization
Abstracting Process Models: Elementary Operations

Elementary aggregation operations

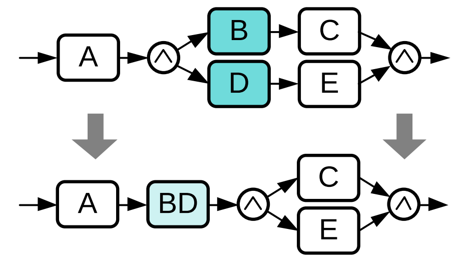
AggrSESE



AggrComplBranches

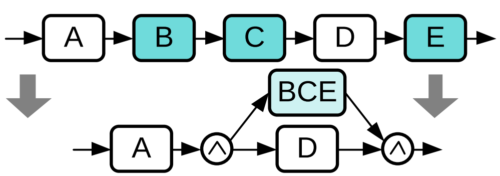


AggrShiftOut

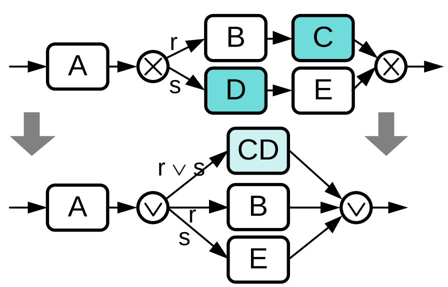


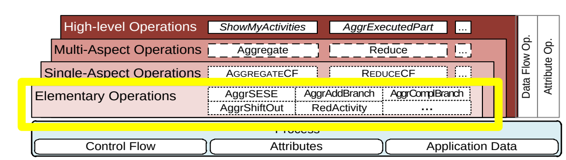
Non-connected activity sets

AggrAddBranch



AggrAddBranch



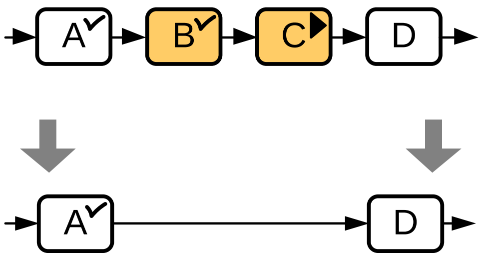


Process Visualization

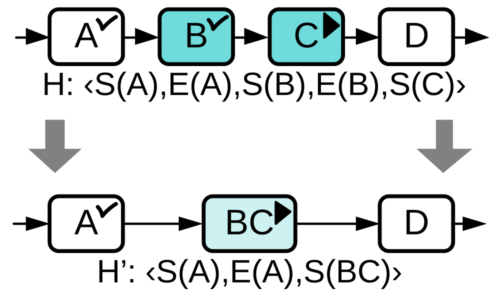
Abstracting Process Models: Elementary Operations

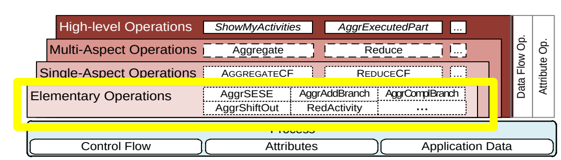
Elementary operations applied at the process instance level

`ReductionSESE`



`AggrSESE`

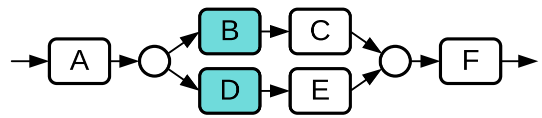




Let’s have a closer look at aggregations!

Abstracting Process Models: View Properties

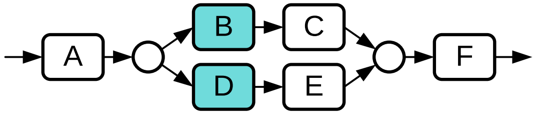
- Example: Aggregate activities from set {B,D}



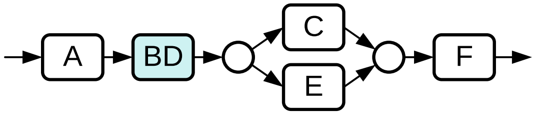
How to aggregate these two activities?

Process Visualization
Abstracting Process Models: View Properties

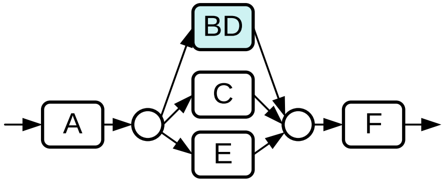
- Example: aggregate activities from set {B,D}



- Alternative 1: `AggrShiftOut`

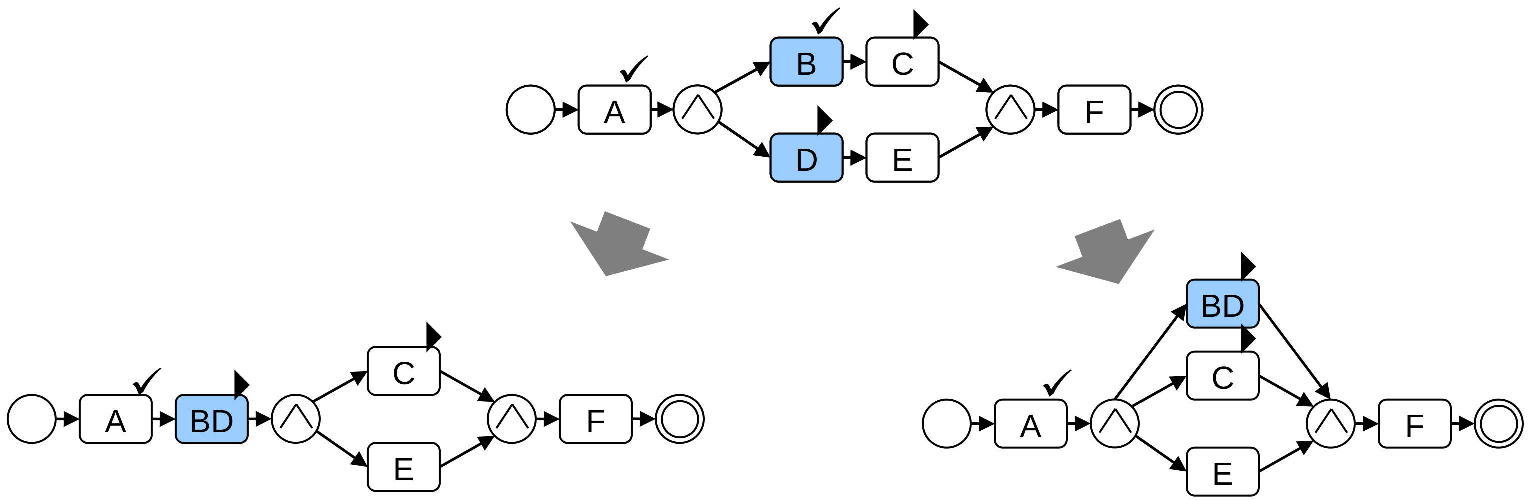


- Alternative 2: `AggrAddBranch`



► Well-defined properties to characterize the resulting view!

Process Visualization
Abstracting Process Models: View Properties



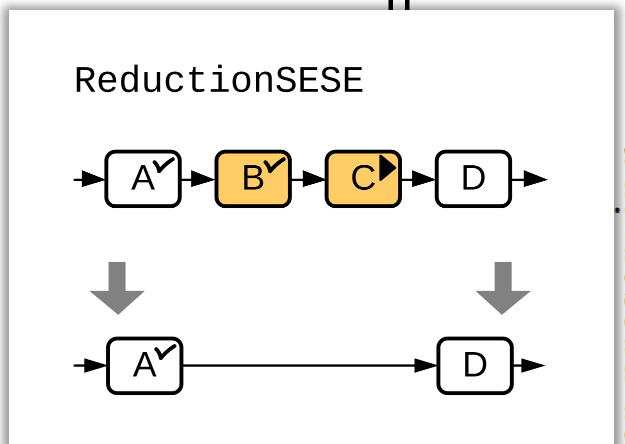
- dependency-generating
- inconsistent process state

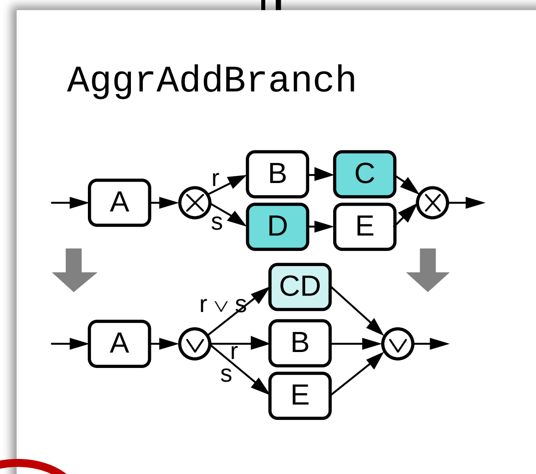
- dependency-erasing
- consistent process state

Abstracting Process Models: View Properties

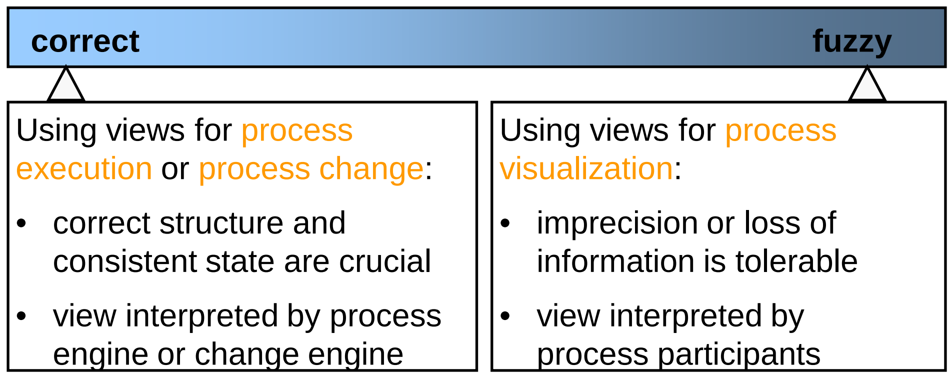
Properties of view-building operations:

	Properties	
Operation	str.	order preserving
RedActivity	-	+
AggrSESE	+	+
AggrComplBranches	+	+
AggrShiftOut	+	+
AggrAddBranch	-	+





Abstracting Process Models: View Parameterization



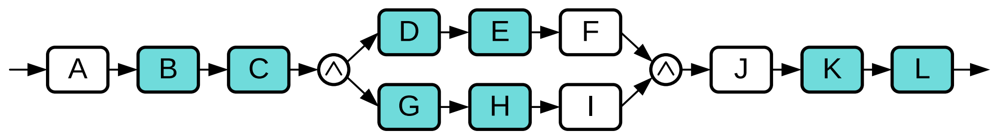
- Proviado addresses this issue by enabling **parameterizable process views**, i.e., the degree of imprecision or tolerable information loss may be flexibly chosen

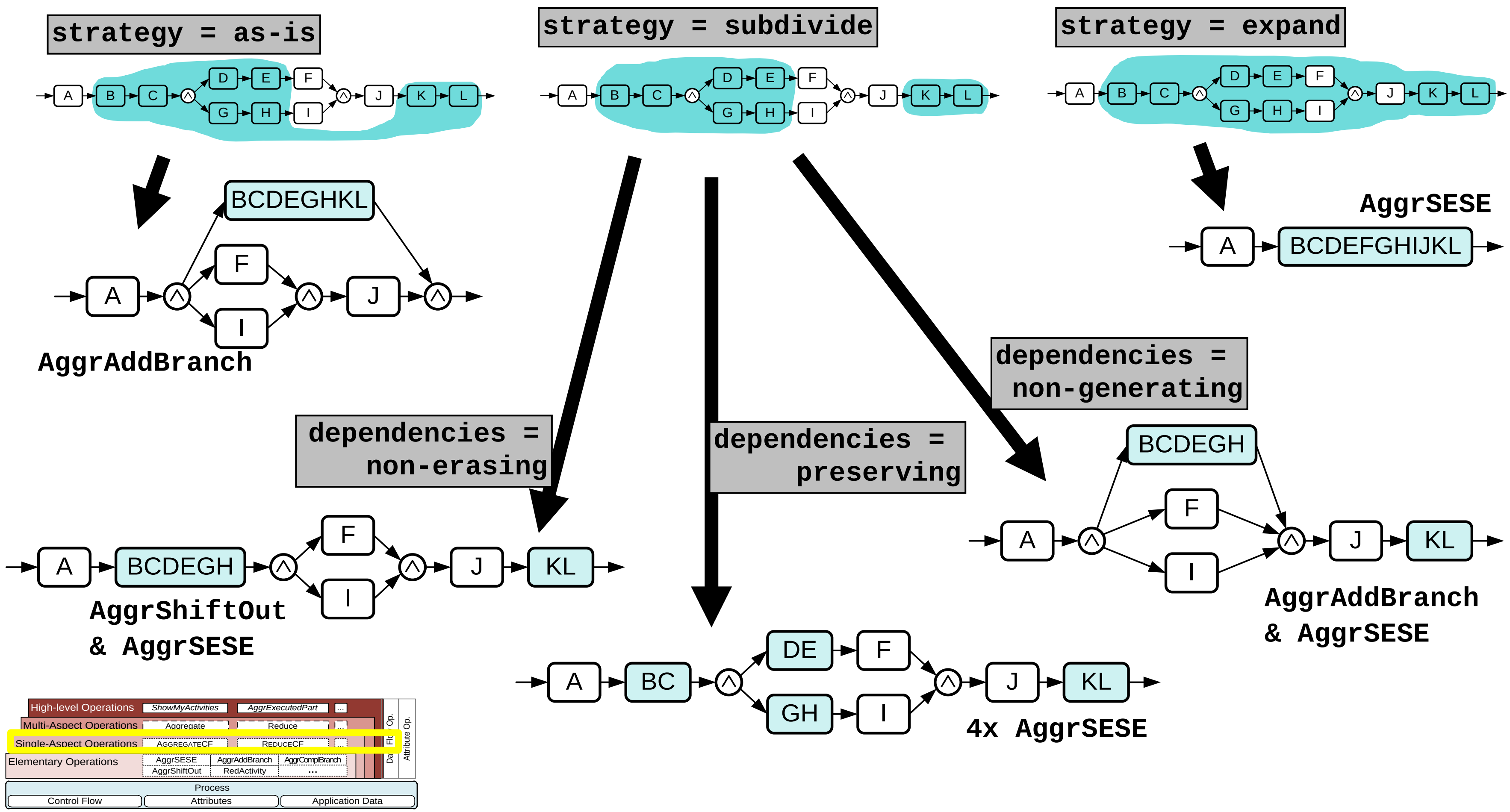
Reichert, Kolb, Bobrik, & Bauer, 2012

Process Visualization

Abstracting Process Models: View Parameterization

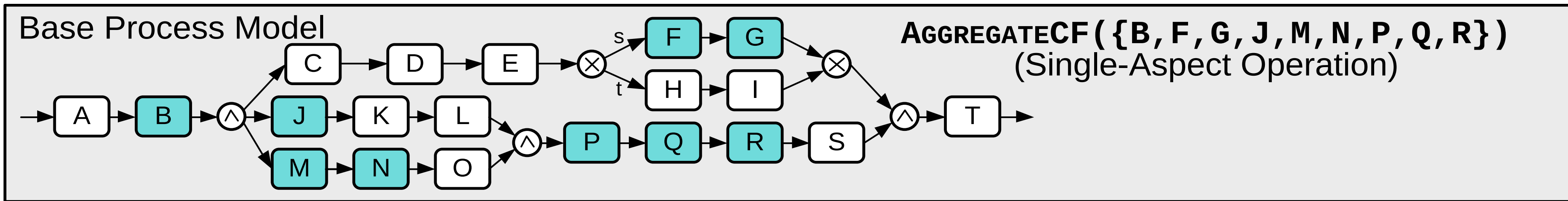
3 strategies for aggregating an activity set (AggregateCF):

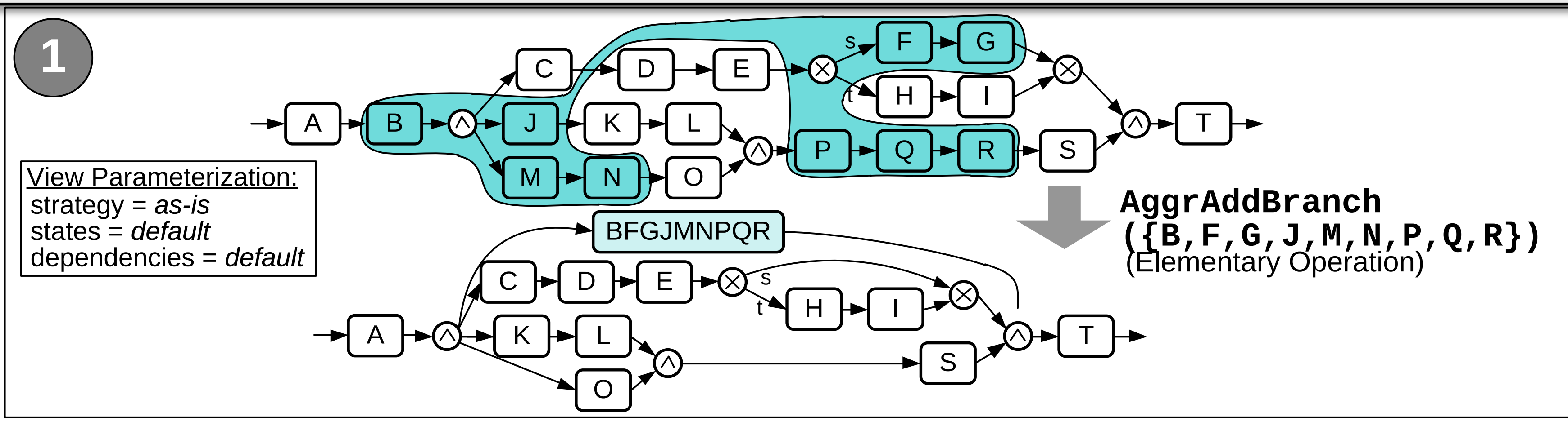




Process Visualization

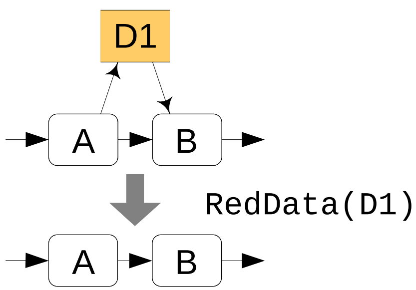
Abstracting Process Models: Single-Aspect Operations

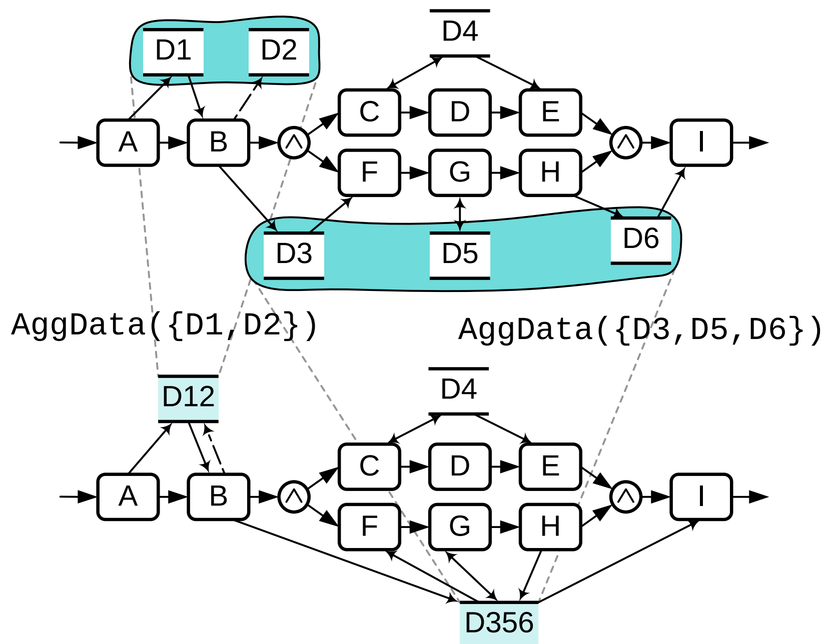




Abstracting Process Models: Other Process Aspects

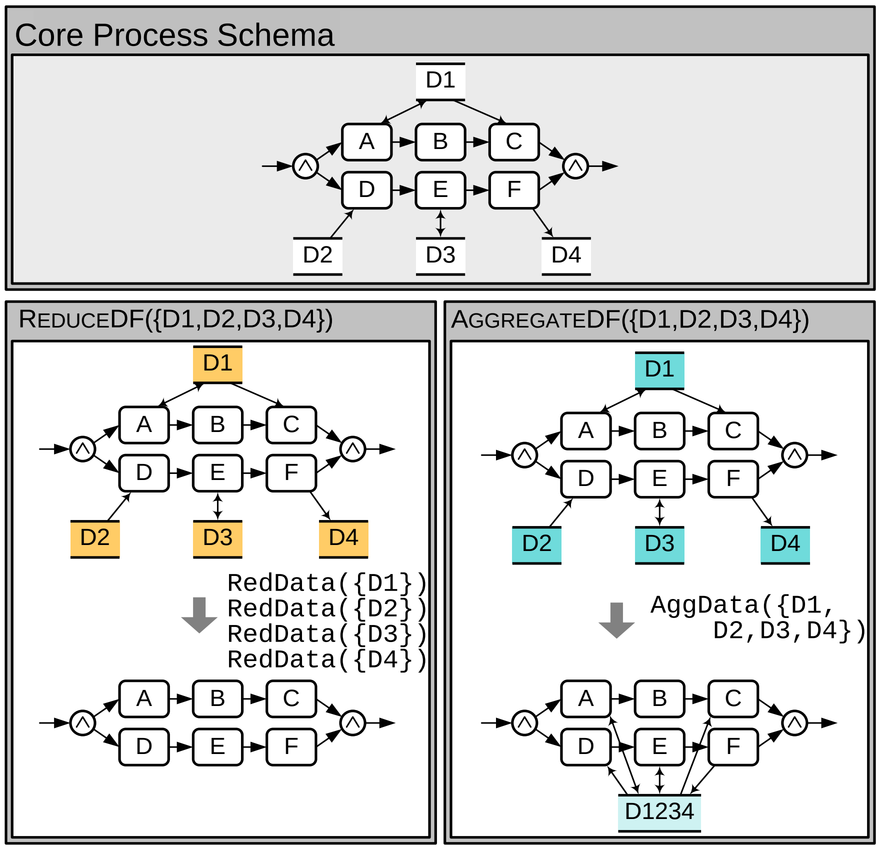
Elementary operations for reducing and aggregating data elements: RedData and AggData





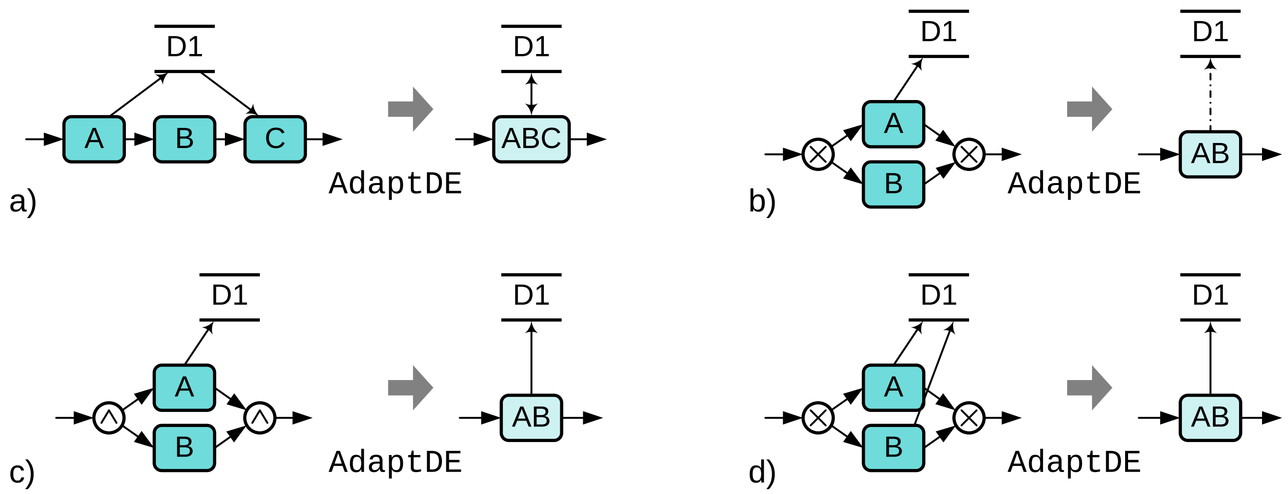
Abstracting Process Models: Other Process Aspects

Single-aspect operations for abstracting data flow:
REDUCEDF and AGGREGATE DF



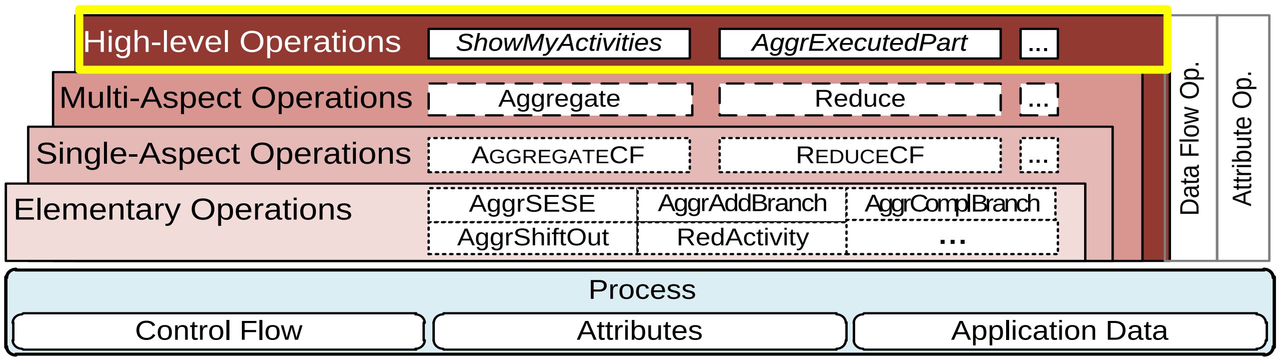
Abstracting Process Models: Other Process Aspects

Adapting data flow edges in the context of control flow aggregations: AdaptDE



Process Visualization
Abstracting Process Models: High-level Operations

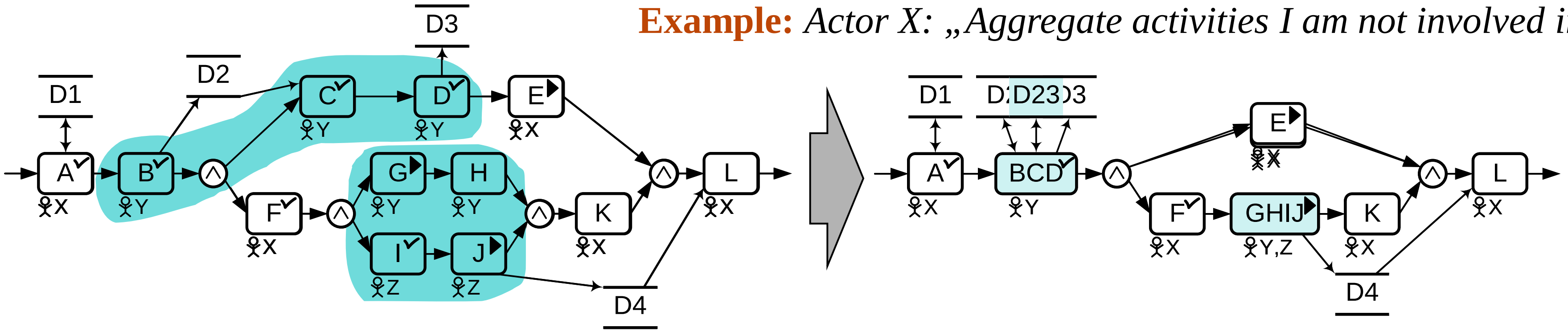
- A user-friendly definition of process views requieres high-level view creation operations



- Example of a high-level operation: *ShowMyActivities*
  - Eliminate all activities the current user is not involved in

Process Visualization
Abstracting Process Models: High-level Operations

Example: *Actor X: „Aggregate activities I am not involved in“*



1. Choose high-level view operation and set its parameters	AggrForeignActivities(*User = X*)
2. Map high-level operation to multi-aspect operation(s)	Aggregate({*B,C,D,G,H,I,J*}, Param = {…})
3. Map multi-aspect operations(s) to single-aspect operations	AGGREGATECF({*B,C,D,G,H,I,J*}, *Param = {strategy = subdivide, …*}) AGGREGATEDF({*D2,D3*})
4. Determine corresponding single-aspect operations and apply them to the original process model; apply refactorings if applicable	AggrShiftOut({*B,C,D*}) AggrSESE({G,H,I,J}) AggrData({D2,D3}, …) AdaptDE({D2}) AdaptDE({D3}) AdaptDE({D4})

Process Visualization

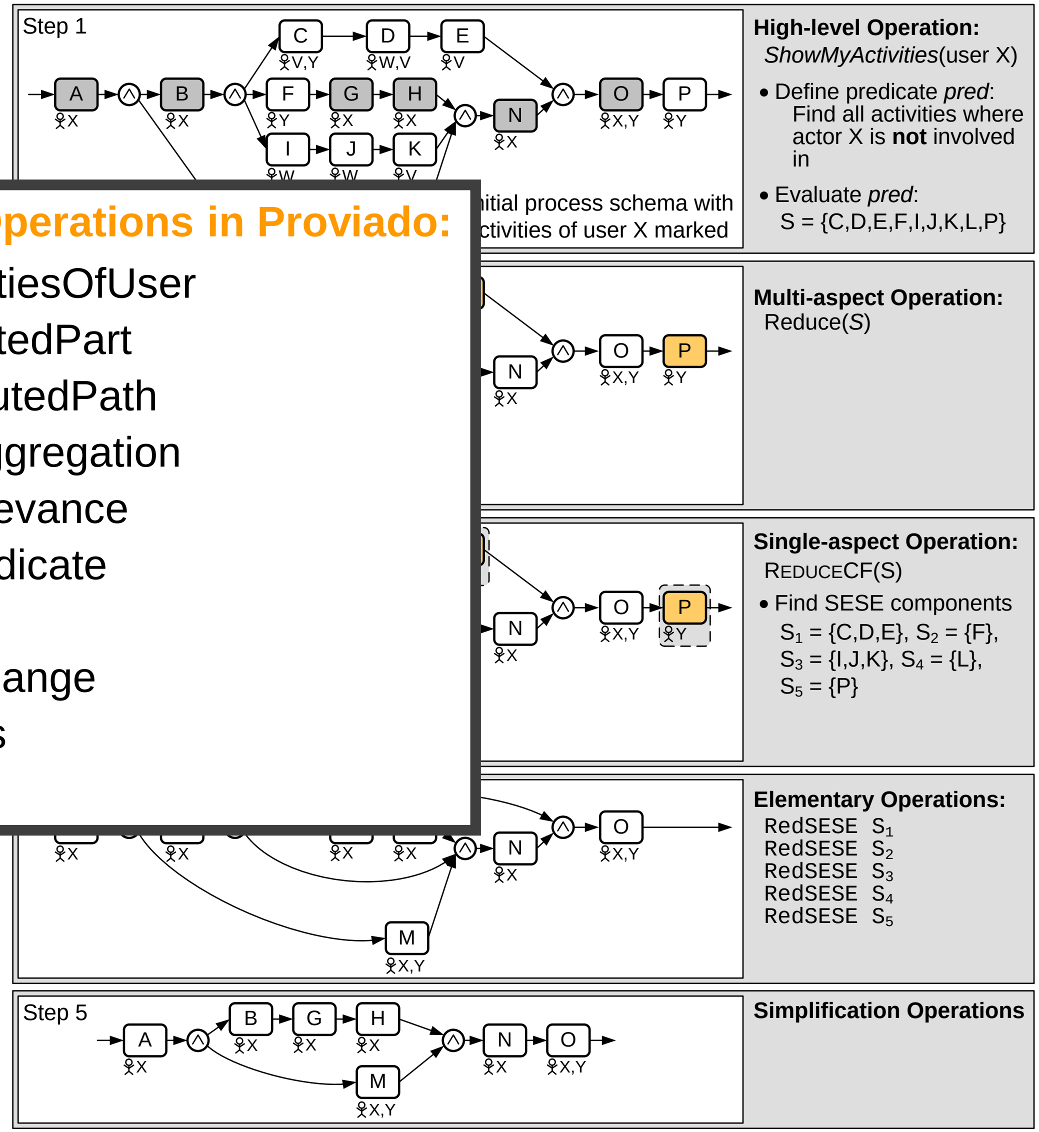
Abstracting Process Models: High-level Operations

Example:

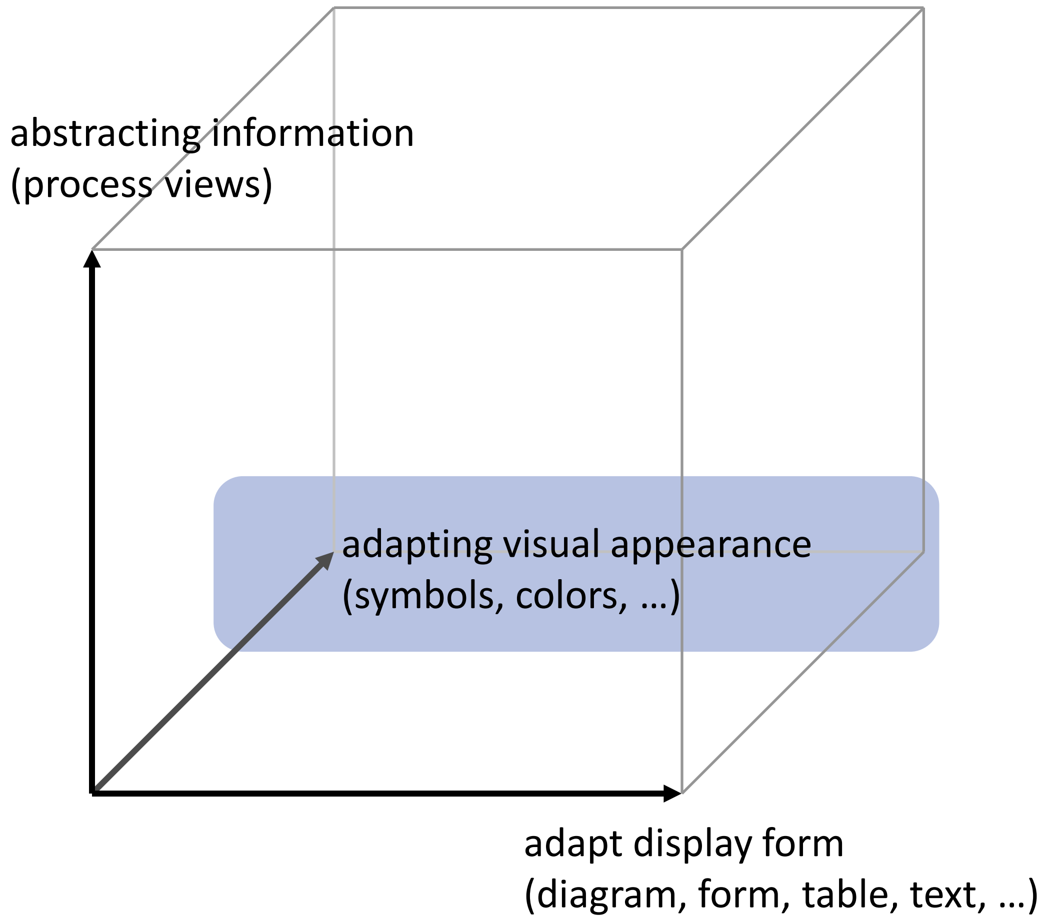
ShowMyActivit

High-level Operations in Proviado:

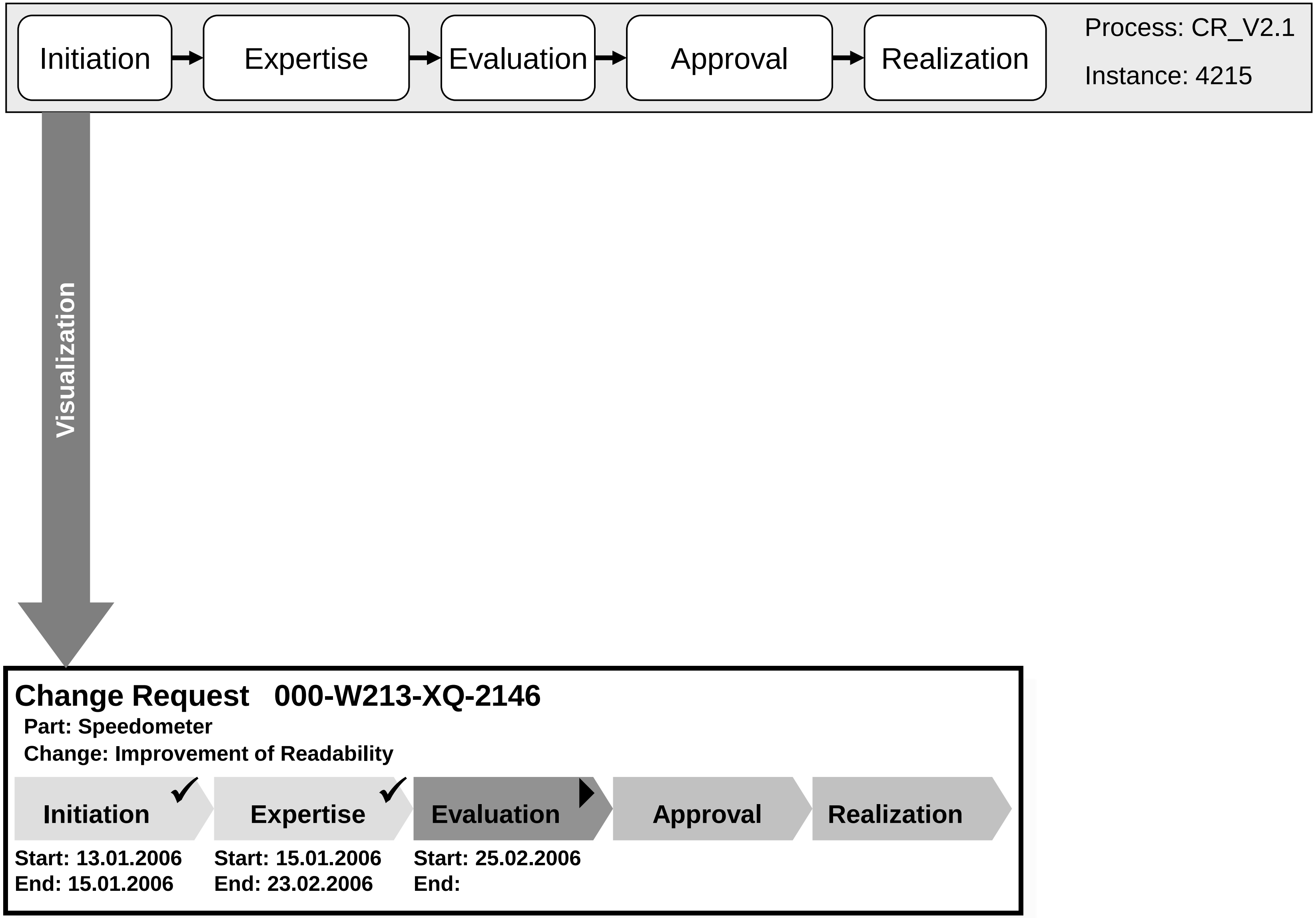
- ShowActivitiesOfUser
- AggrExecutedPart
- ShowExecutedPath
- GroupedAggregation
- ViewByRelevance
- ViewByPredicate
- Subgraph
- SubgraphRange
- CutProcess
- …



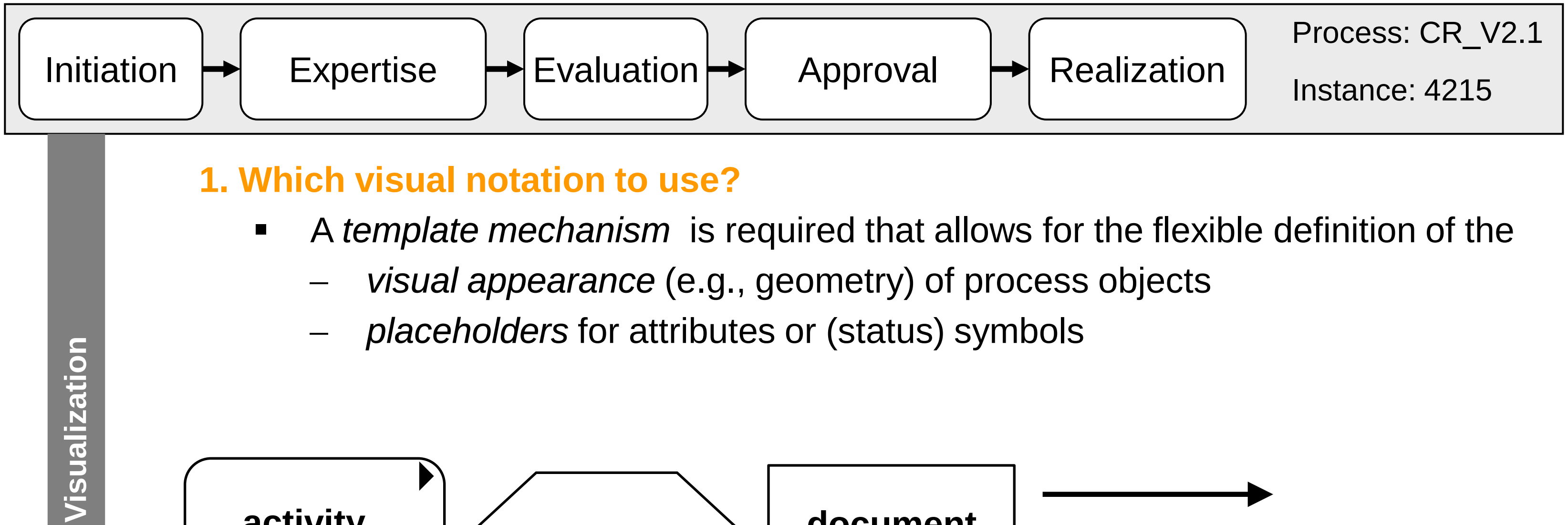
Process Visualization Dimensions



Process Visualization
Visual Appearance of a Process Model: Issues

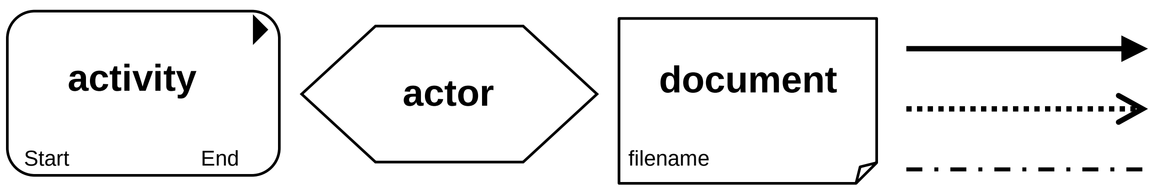


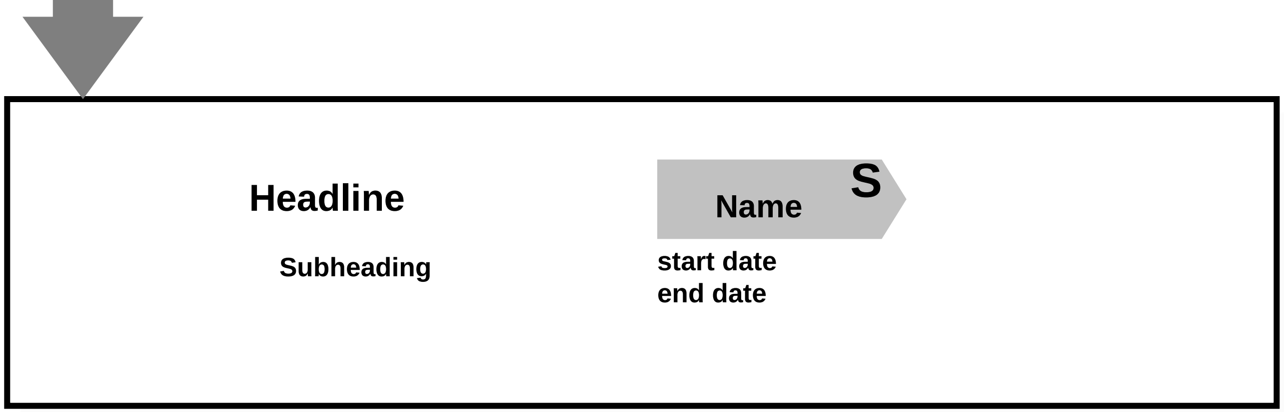
Process Visualization
Visual Appearance of a Process Model: Issues



1. Which visual notation to use?

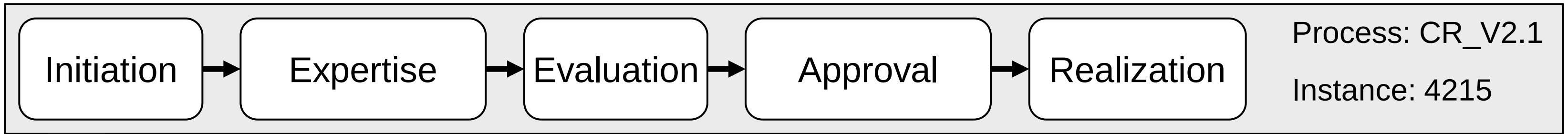
- A *template mechanism* is required that allows for the flexible definition of the
  - *visual appearance* (e.g., geometry) of process objects
  - *placeholders* for attributes or (status) symbols





Process Visualization

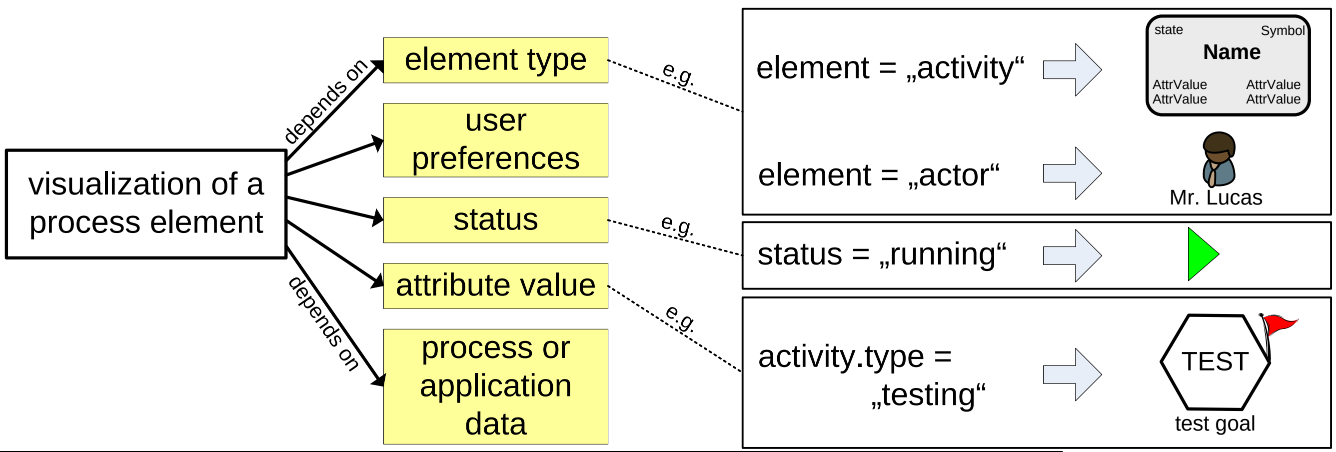
Visual Appearance of a Process Model: Issues

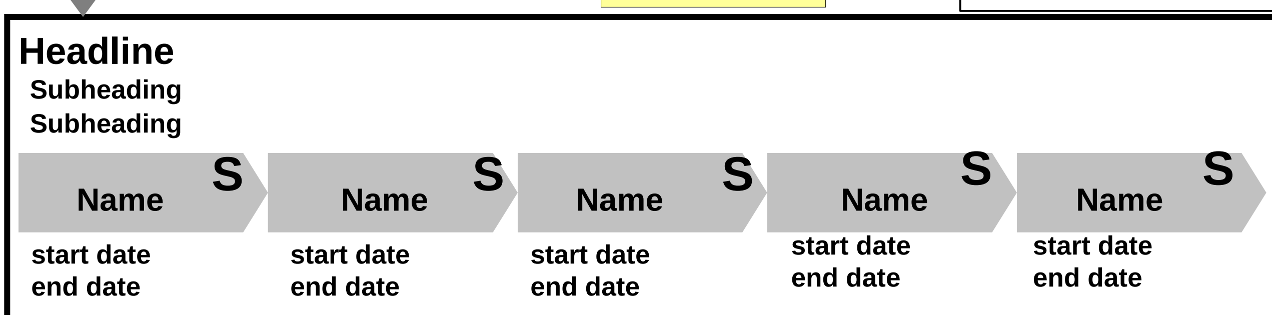


1. Which visual notation to use?

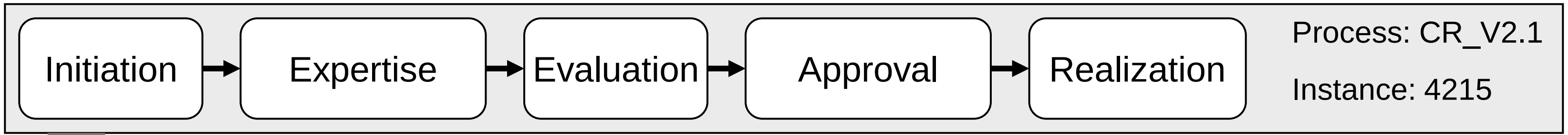
2. When to use which visualization?

- Selection of a concrete *visualization template* may depend on the type of a process element, attributes, process data, users, etc.





Process Visualization
Visual Appearance of a Process Model: Issues

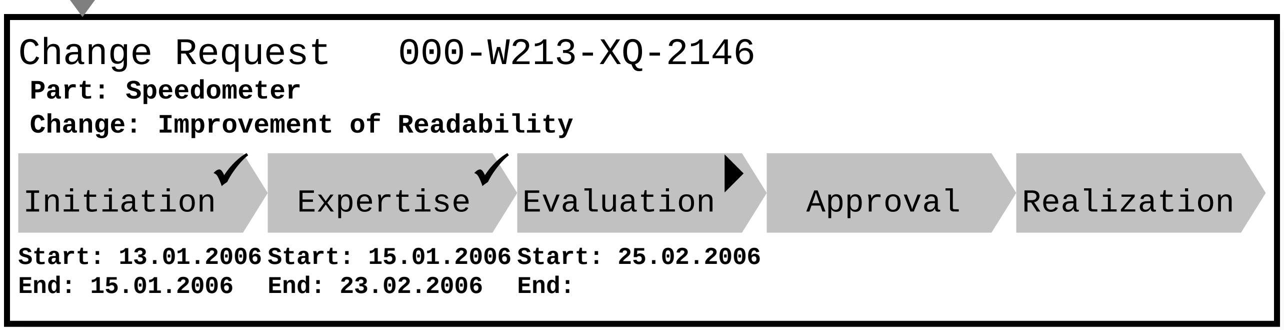


1. Which visual notation to use?

2. When to use which visualization?

3. Which data? Where to add and how?

- Abstracting and formatting process data
  - Formatting, e.g., date values
  - Abstracting, e.g. costs: "high " instead of 1.000.000 €
- Filling template parameters with concrete process data



Process Visualization

Visual Appearance of a Process Model: Issues

Initiation → Expertise → Evaluation → Approval → Realization

Process: CR_V2.1

Instance: 4215

Visualization

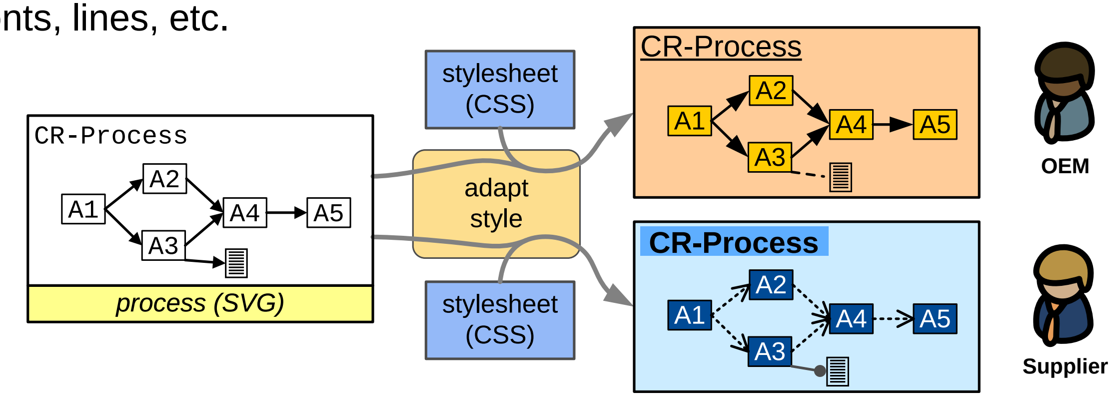
1. Which visual notation to use?

2. When to use which visualization?

3. Which data? Where to add and how?

4. How shall the visualization style look like?

- Colors, fonts, lines, etc.



Change Request 000-W213-XQ-2146

Part: Speedometer

Change: Improvement of Readability

Initiation ✓	Expertise ✓	Evaluation	Approval	Realization
Start: 13.01.2006 End: 15.01.2006	Start: 15.01.2006 End: 23.02.2006	Start: 25.02.2006 End:		

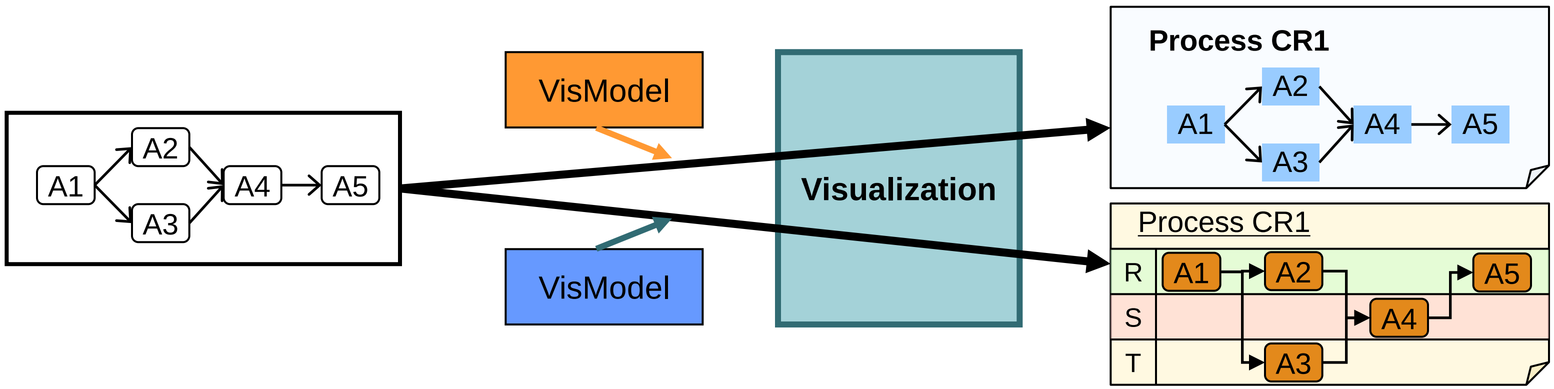
Process Visualization

Visual Appearance of a Process Model: The Proviado Approach

How we addressed these issues in the Proviado visualization framework …

Visual Appearance of a Process Model: The Proviado Approach

Basic principle: separate process data from its presentation

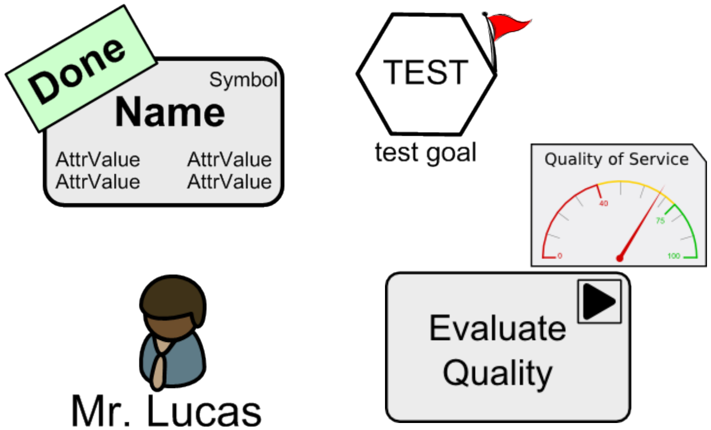


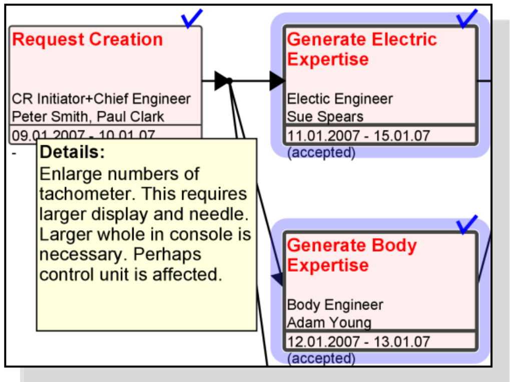
Visualization Model

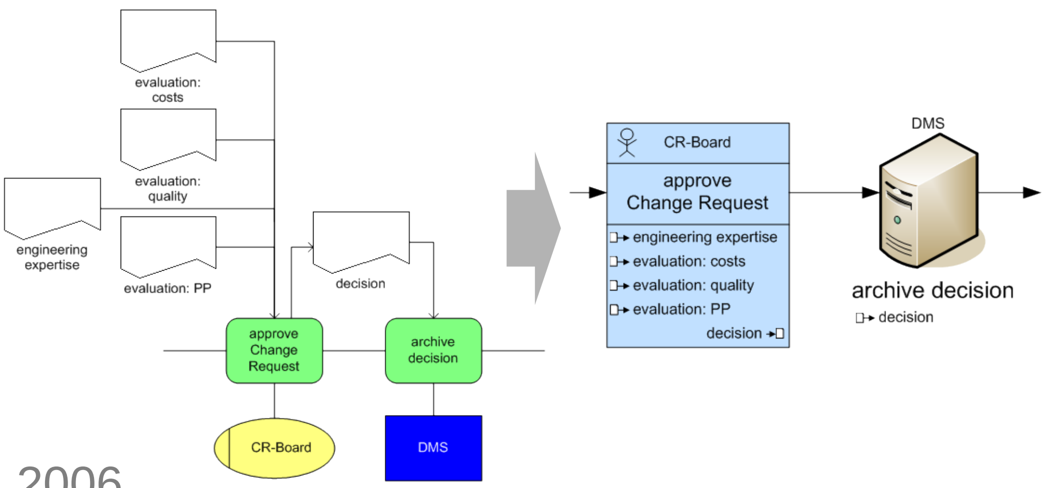
- logical description of all information required for creating a particular process visualization, e.g., symbols, display form, layouting, …

Visual Appearance of a Process Model: The Proviado Approach

Examples of visualization templates







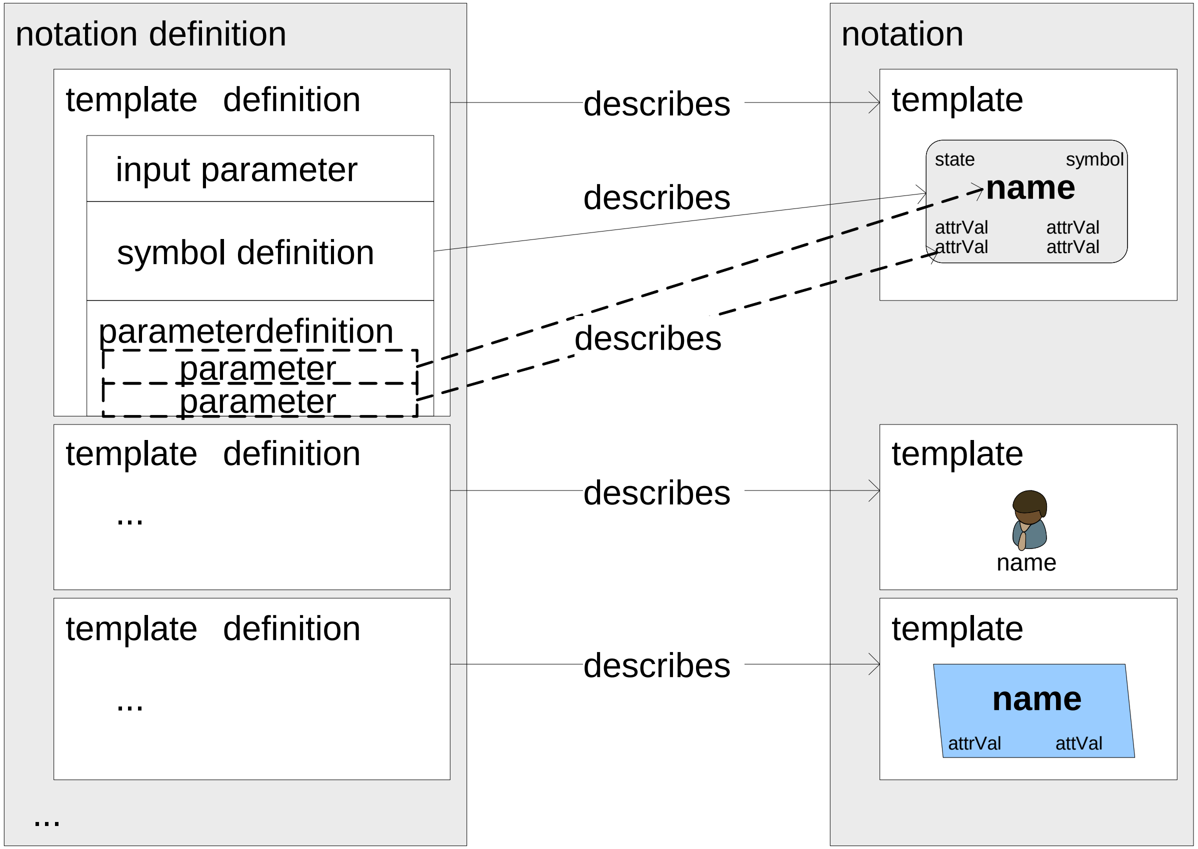
Visualization template defines

1. symbol to be used
2. data to be displayed
3. application context

Bobrik et al., 2006

Visual Appearance of a Process Model: The Proviado Approach

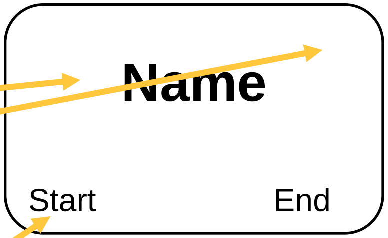
Schema of a visualization templates



Visual Appearance of a Process Model: The Proviado Approach

Defining a visualization templates for process elements

```
<template id="default_act">
  <inputs>...</inputs>
  <graphic>
    <symbol>...</symbol>
    <parameter name="Beschreibung"
      location='g/text[@name='actname']"
      value="act.getName()" />
    <parameter name="Status"
      location='g/g[@name='status']">
      <choice>
        <when test="act.getState()=RUNNING">
          <use xlink:href="#act_state_RUNNING" />
        </when>
        <when test="act.getState()=COMPLETED">...</when>
        ...
      </choice>
    </parameter>
    <parameter name=starttime"
      location='g/text/[@name='starttime']'
      value="formatDate(act.getStart(), 'dd/mm/yyyy')"/>
    ...
  </graphic>
</template>
```



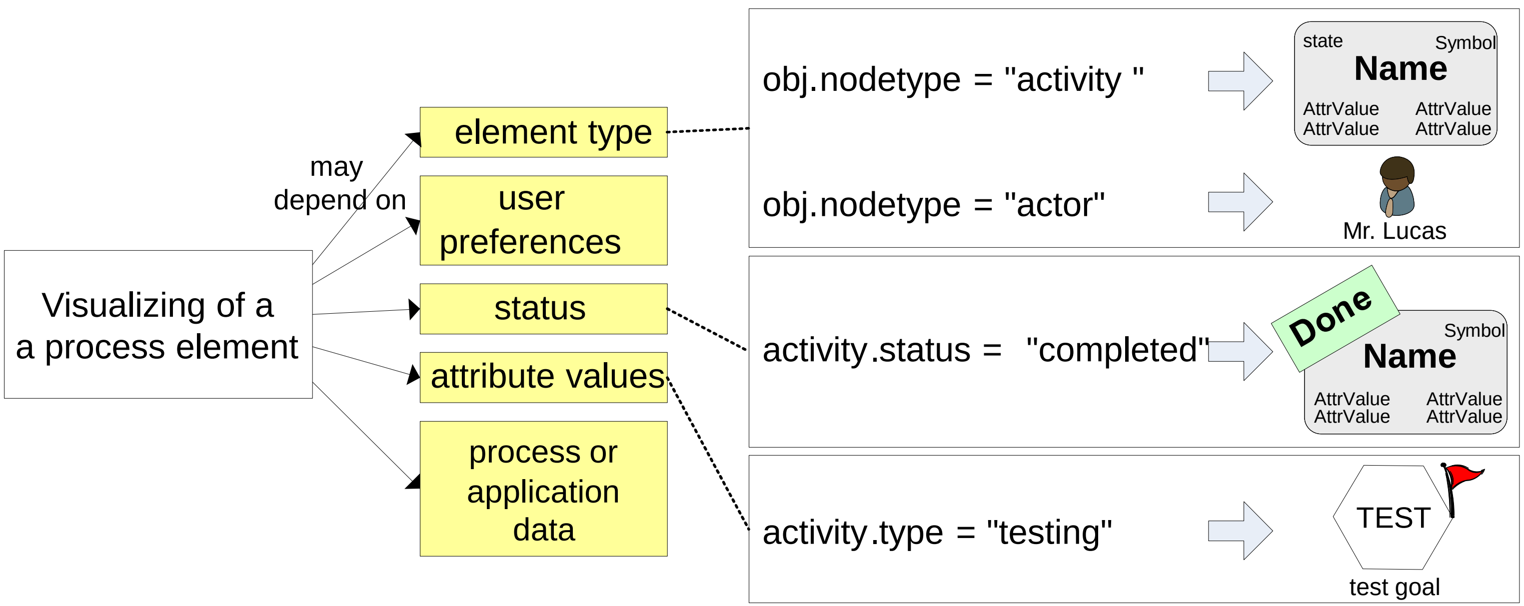
Symbol shapes are defined using SVG

Parameter locations inside the symbol are specified using XPath

JavaScript can be used for calculating parameter values

Visual Appearance of a Process Model: The Proviado Approach

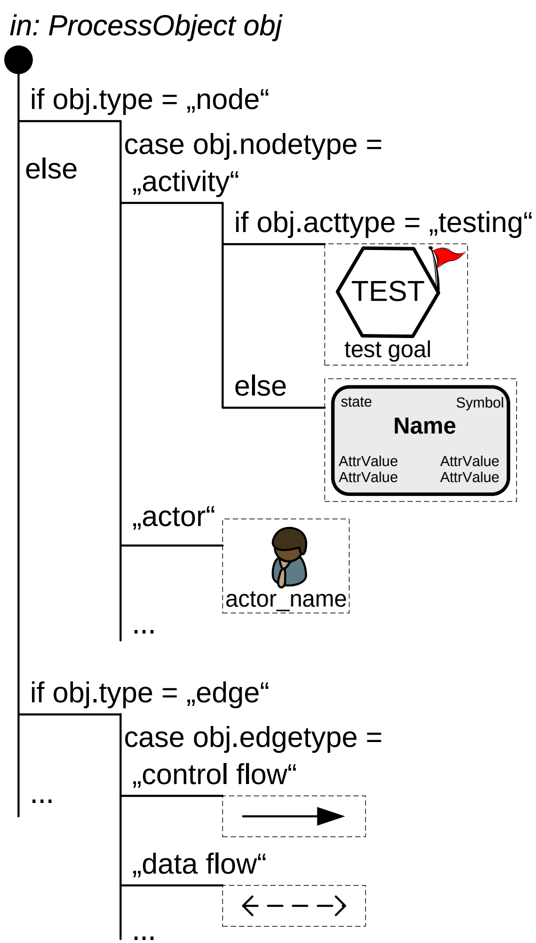
Defining the application context of visualization templates



context rules

Visual Appearance of a Process Model: The Proviado Approach

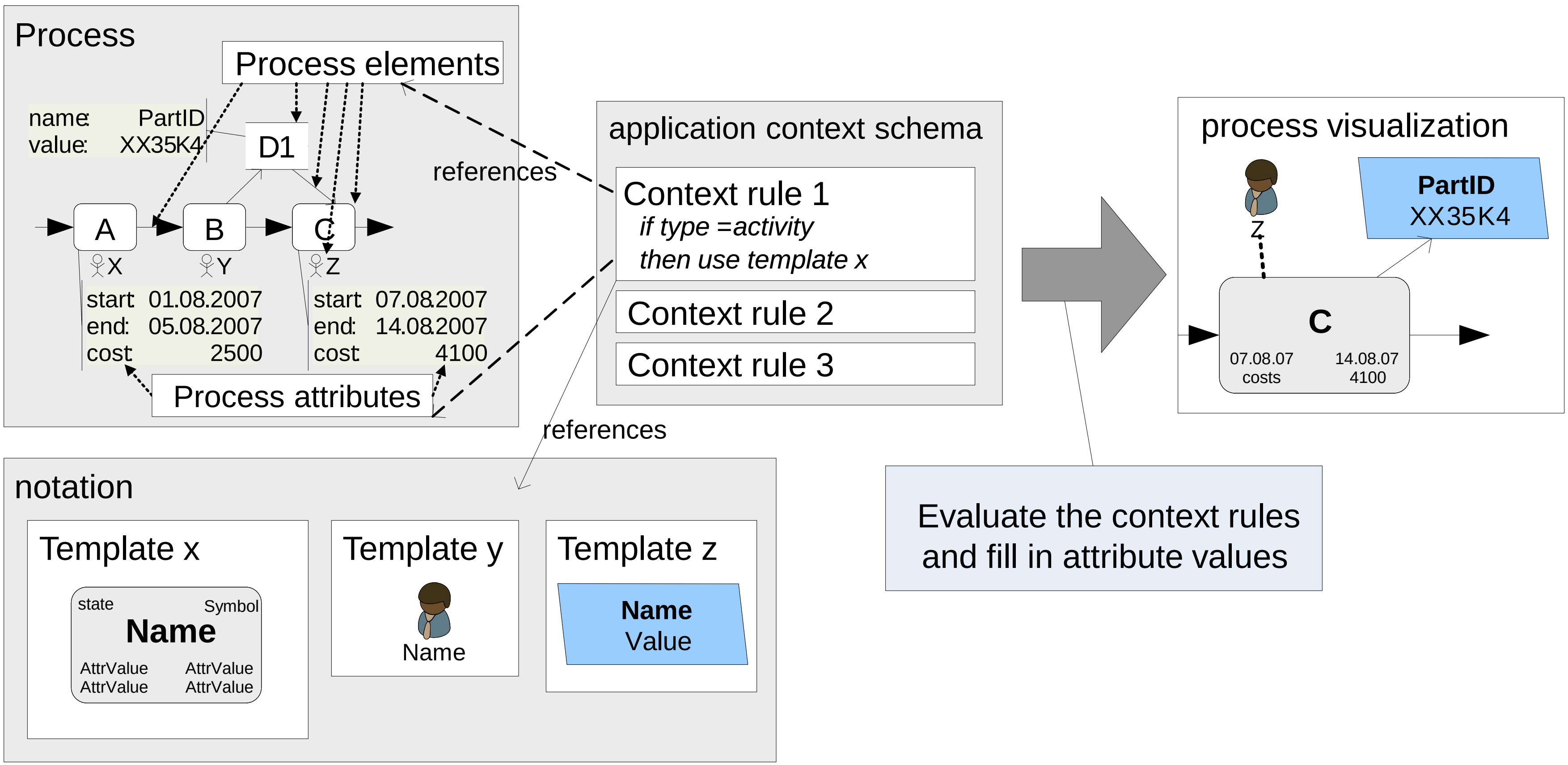
Defining the application context of visualization templates




```
<if test="self.type=ACTOR">
  <template id="actor">
    <inputs>
      <input name="actor" value="self"/>
    </inputs>
  </template>
</if>
<if test="self.type=ACTIVITY">
  <choose>
    <case test="self.type='testing'">
      <template ref="testing_act">
        <inputs>
          <input name="act" value="self"/>
        </inputs>
      </template>
    </case>
    <otherwise>
      <template ref="default_act">
        <inputs>
          <input name="act" value="self"/>
        </inputs>
      </template>
    </otherwise>
  </choose>
</if>
```

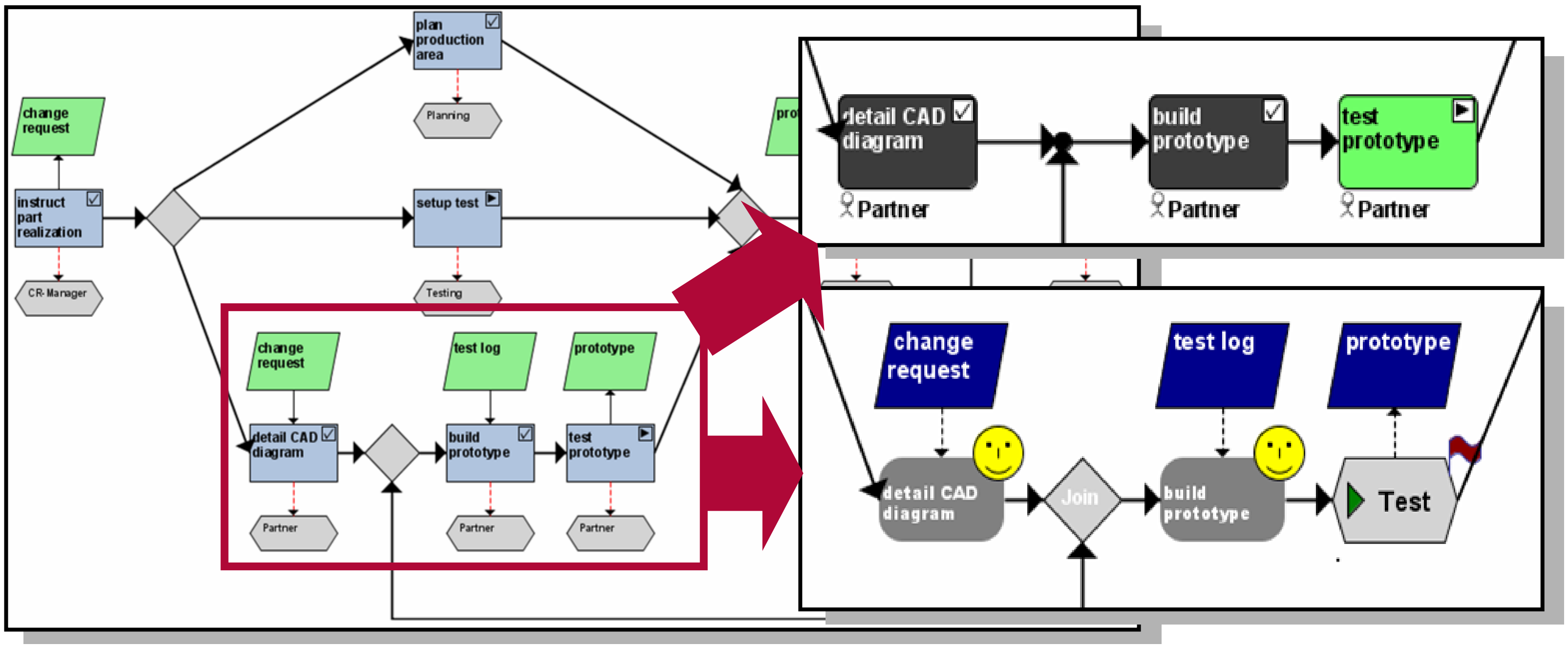
Visual Appearance of a Process Model: The Proviado Approach

Creating a process visualization

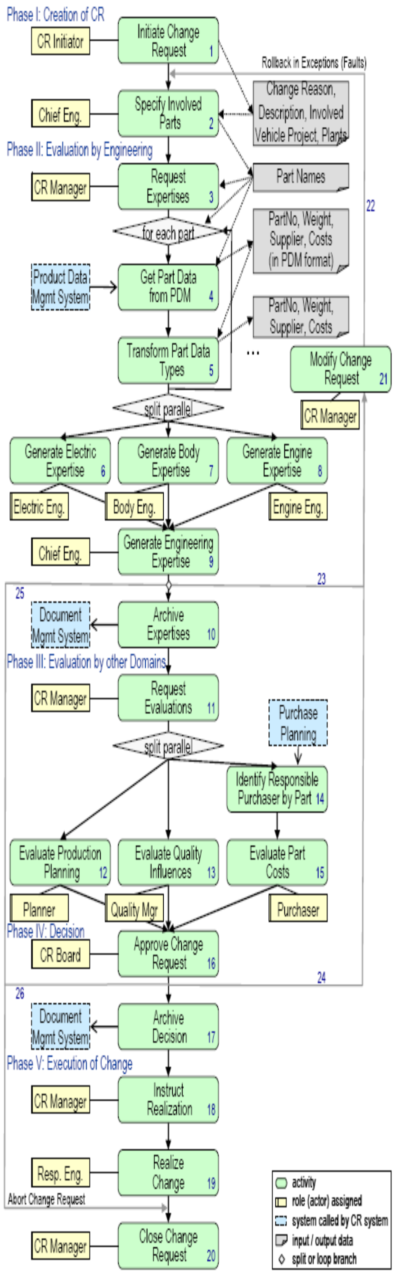


Visual Appearance of a Process Model: The Proviado Approach

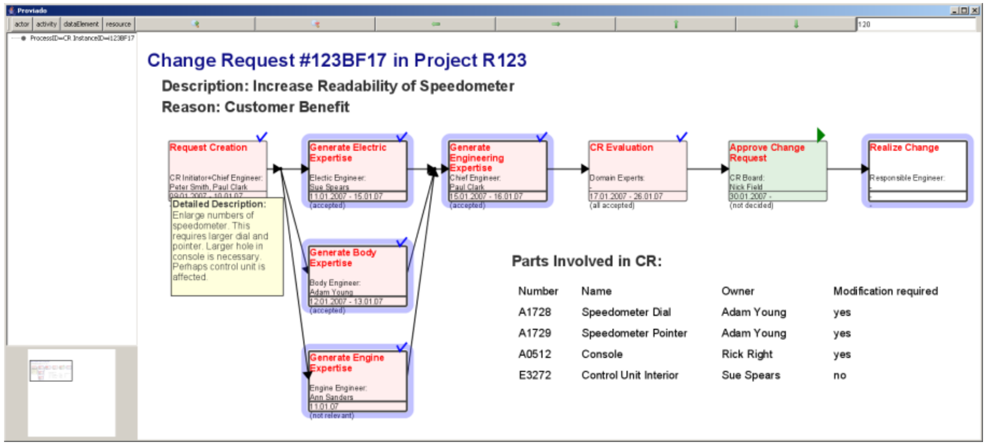
Example



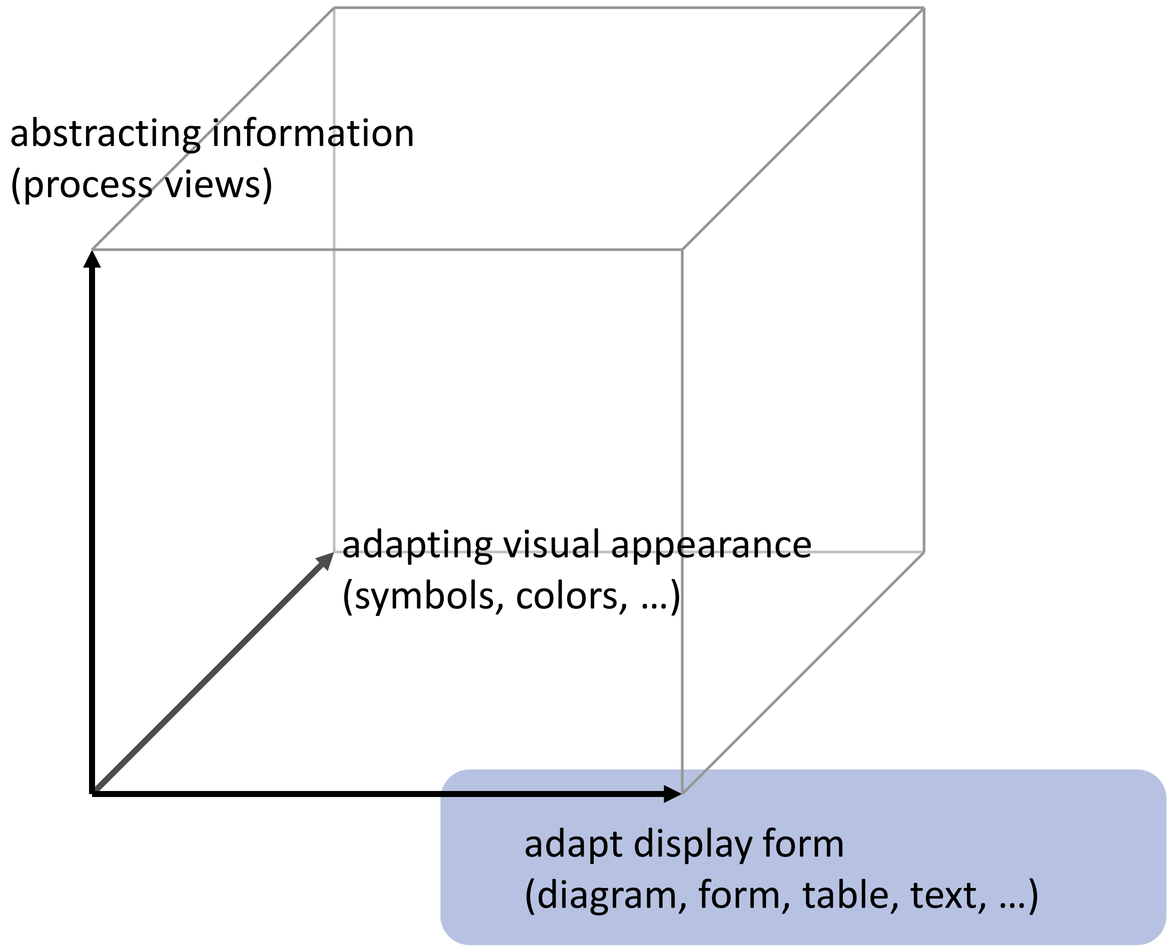
Process Visualization
Combining the Dimensions



Personalized Visualization

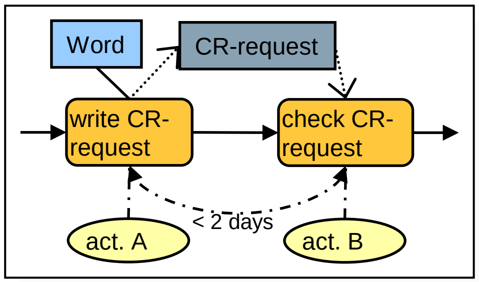


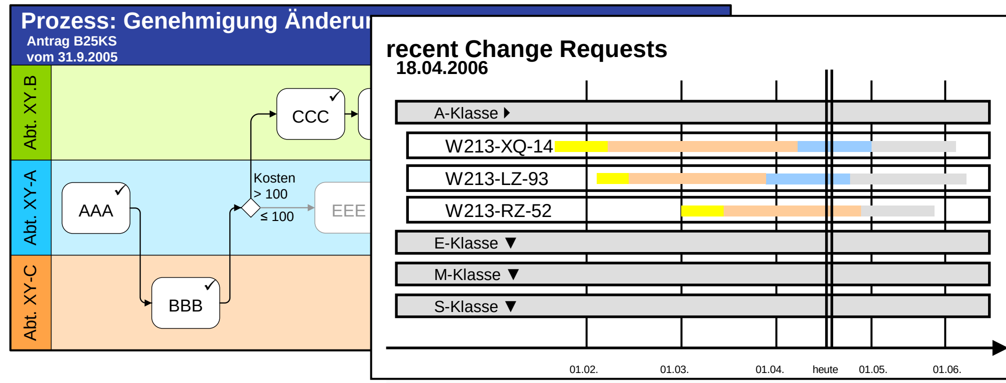
Process Visualization Dimensions

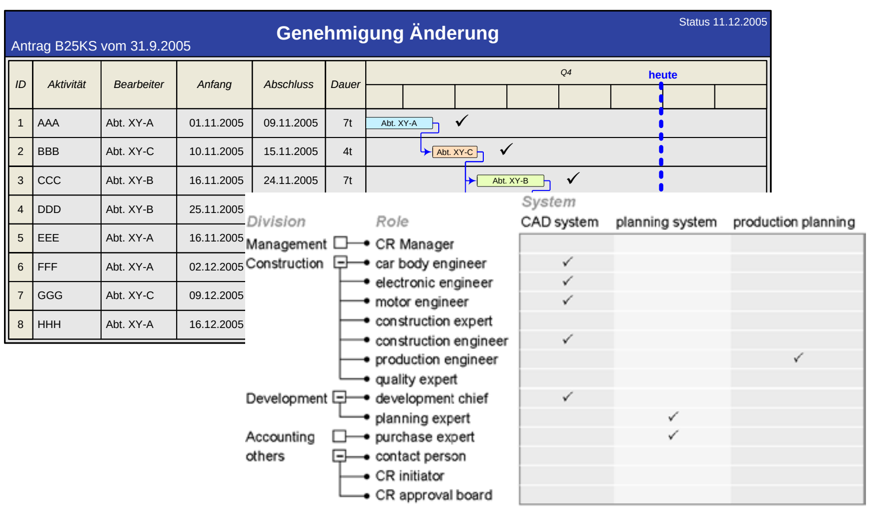


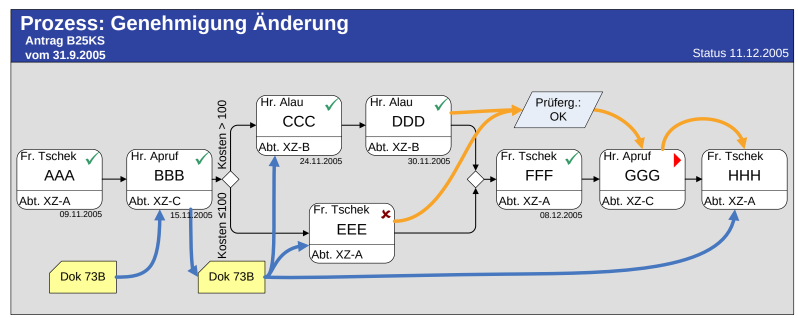
Process Visualization
Display Form of a Process Model

Goal: Experiment with other ways of displaying processes



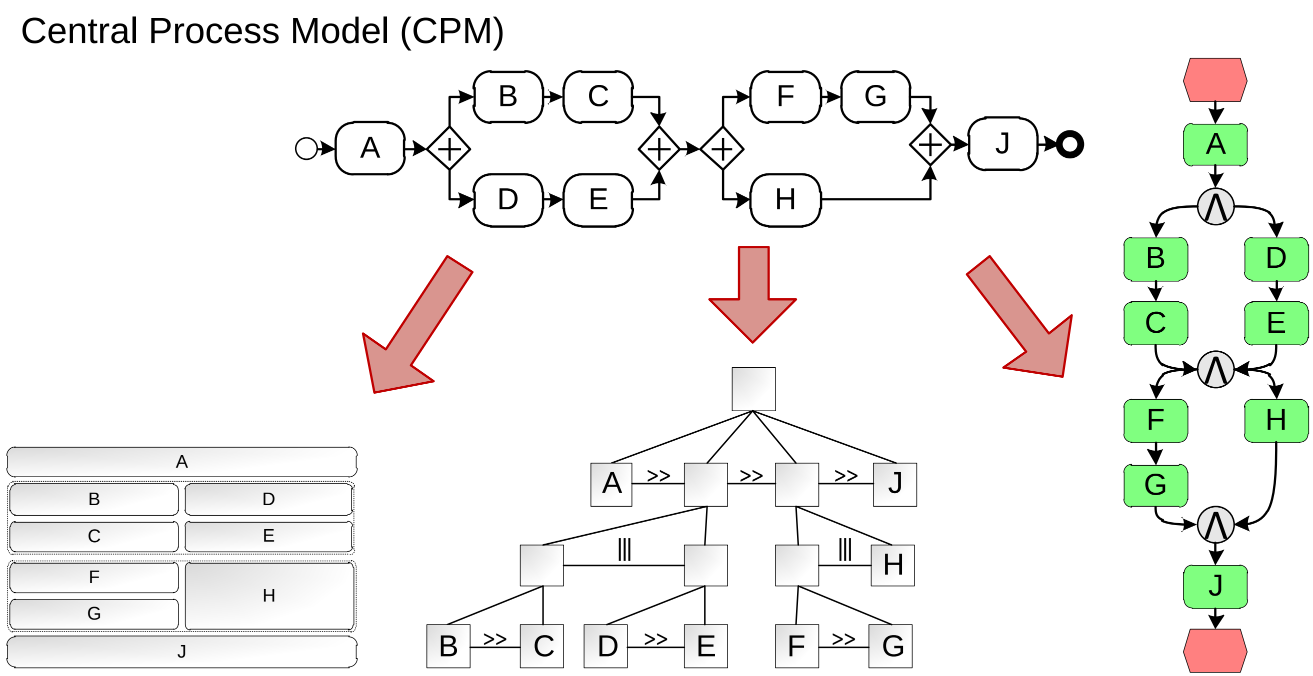






Process Visualization
Display Form of a Process Model

Some concrete work we did in the proView project



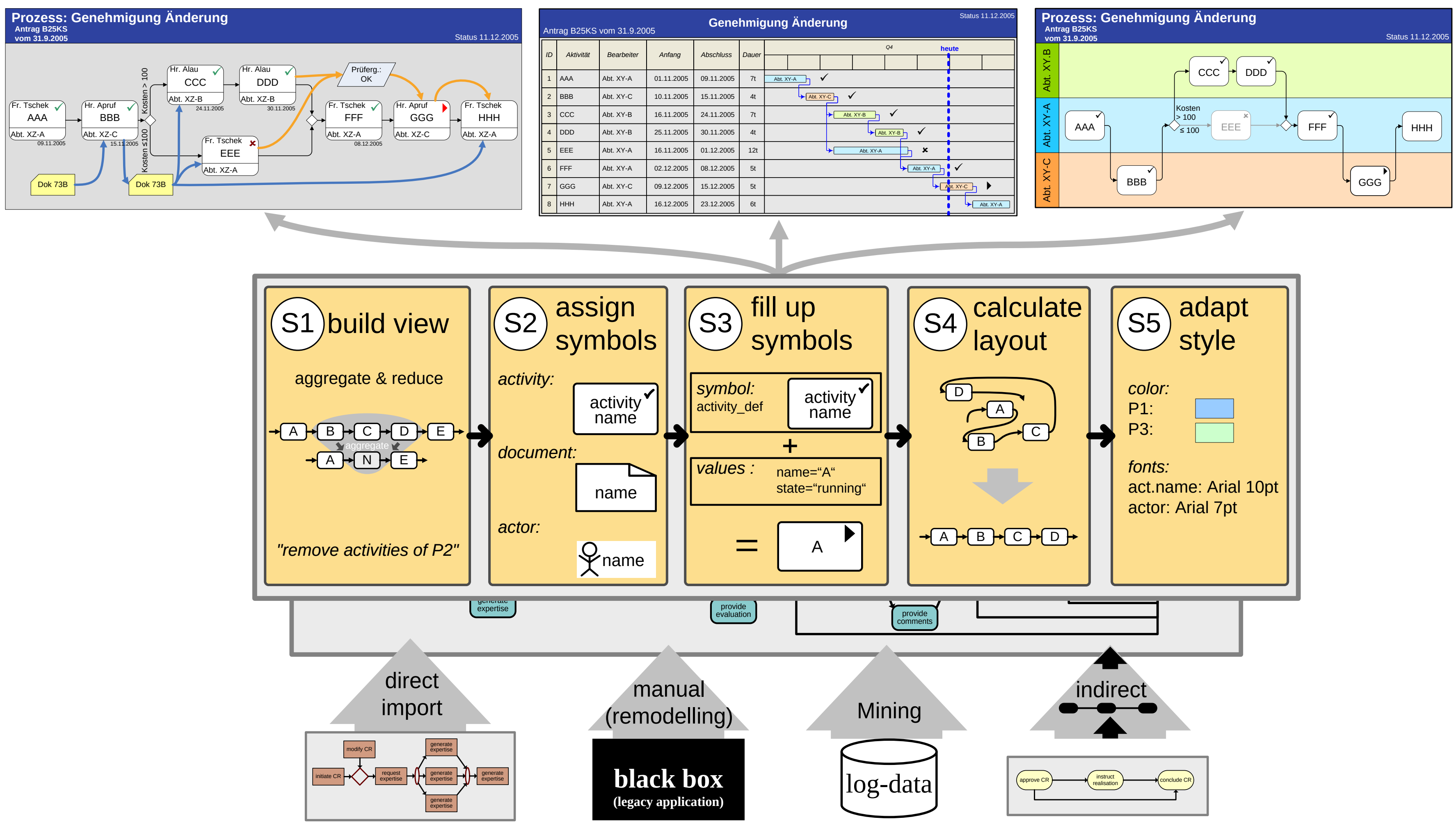
a) Form-based View

b) Tree-based View

c) EPC-based View

Process Visualization

Summary: What we achieved in Proviado?



Updatable Process Model Abstractions

- Focus of this presentation has been on the personalized visualization of large process models
- Another fundamental issue: How to enable domain experts to change large process models!

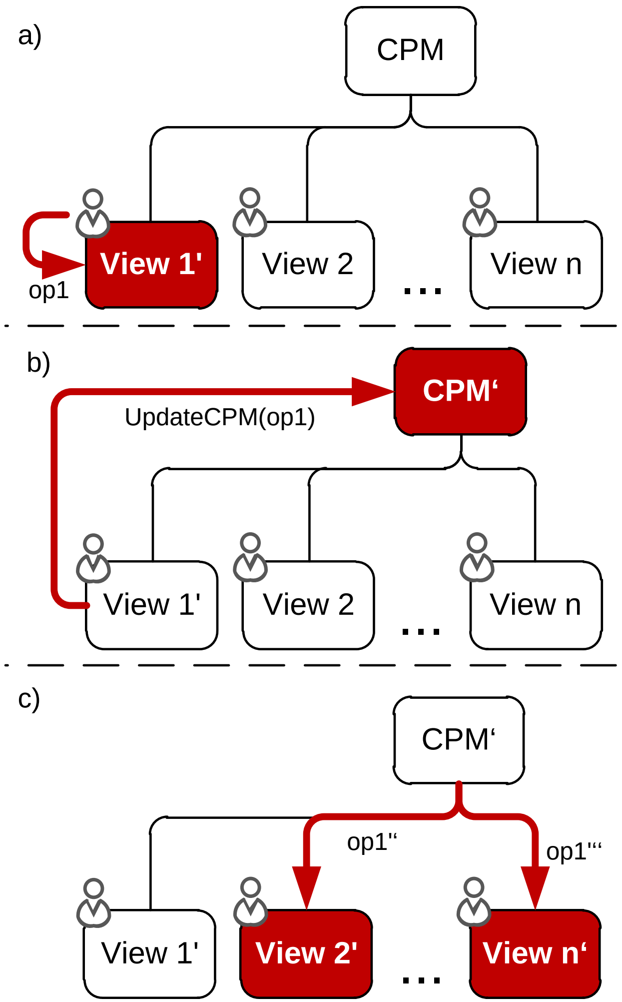
Proviado → proView

Create Update Execute Appearance

Updatable Process Model Abstractions

proView

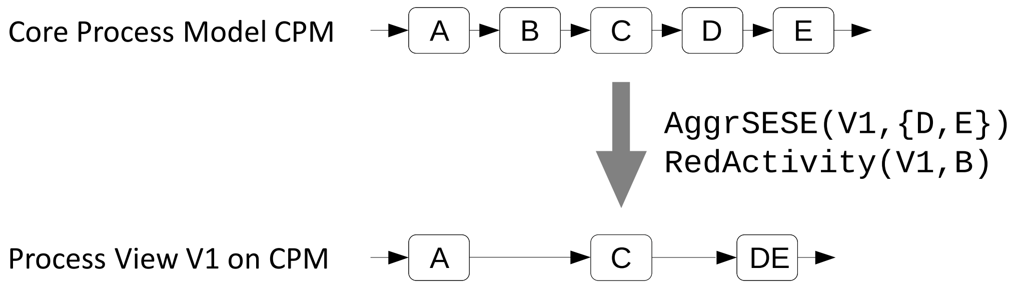
- Basic Idea: Using process views not only for visualization purpose, but also as interface for changing the underlying core process model (CPM)
- Updates of a process view then have to be correctly propagated to its CPM as well as all other views on this CPM
- Necessitates a formal foundation

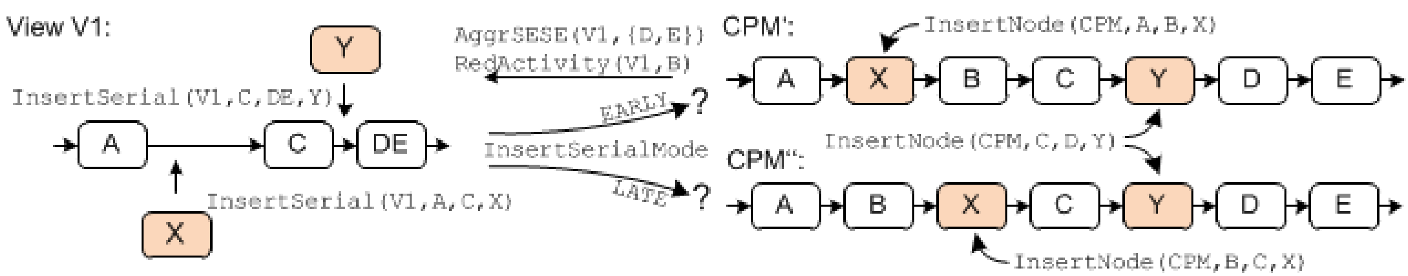


Kolb, Kammerer & Reichert, 2012

Updatable Process Model Abstractions

proView



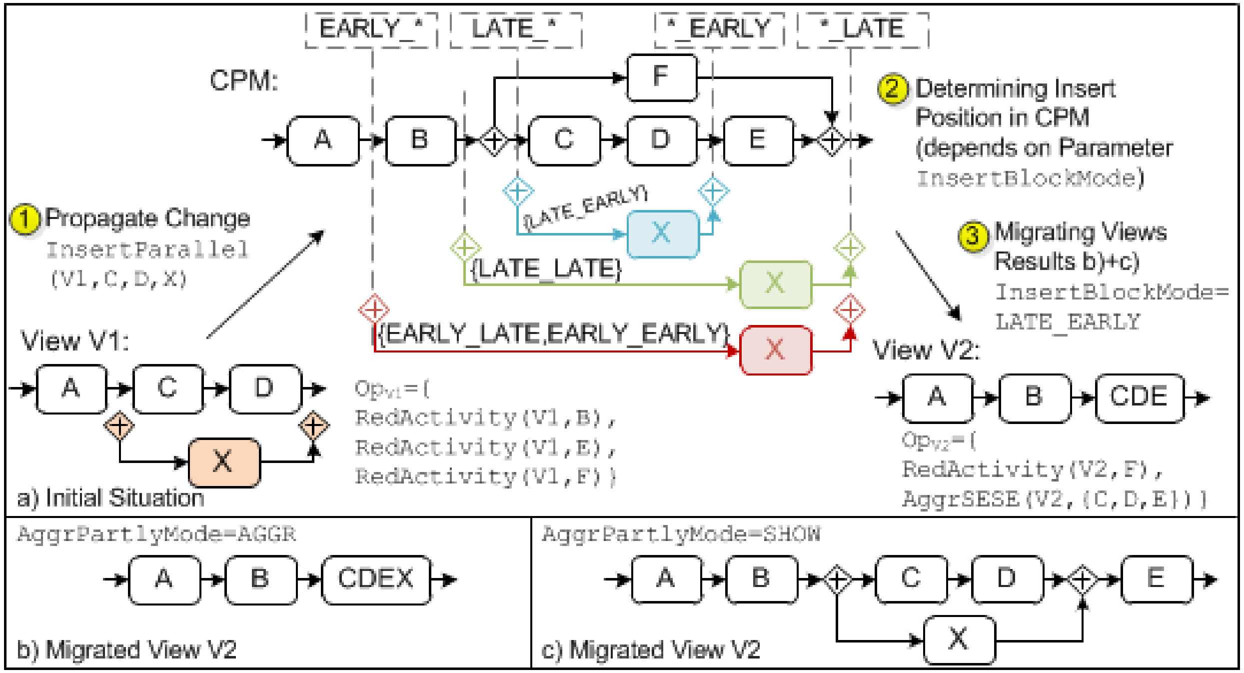


Ambiguities when propagating view changes to the CPM

Updatable Process Model Abstractions

proView

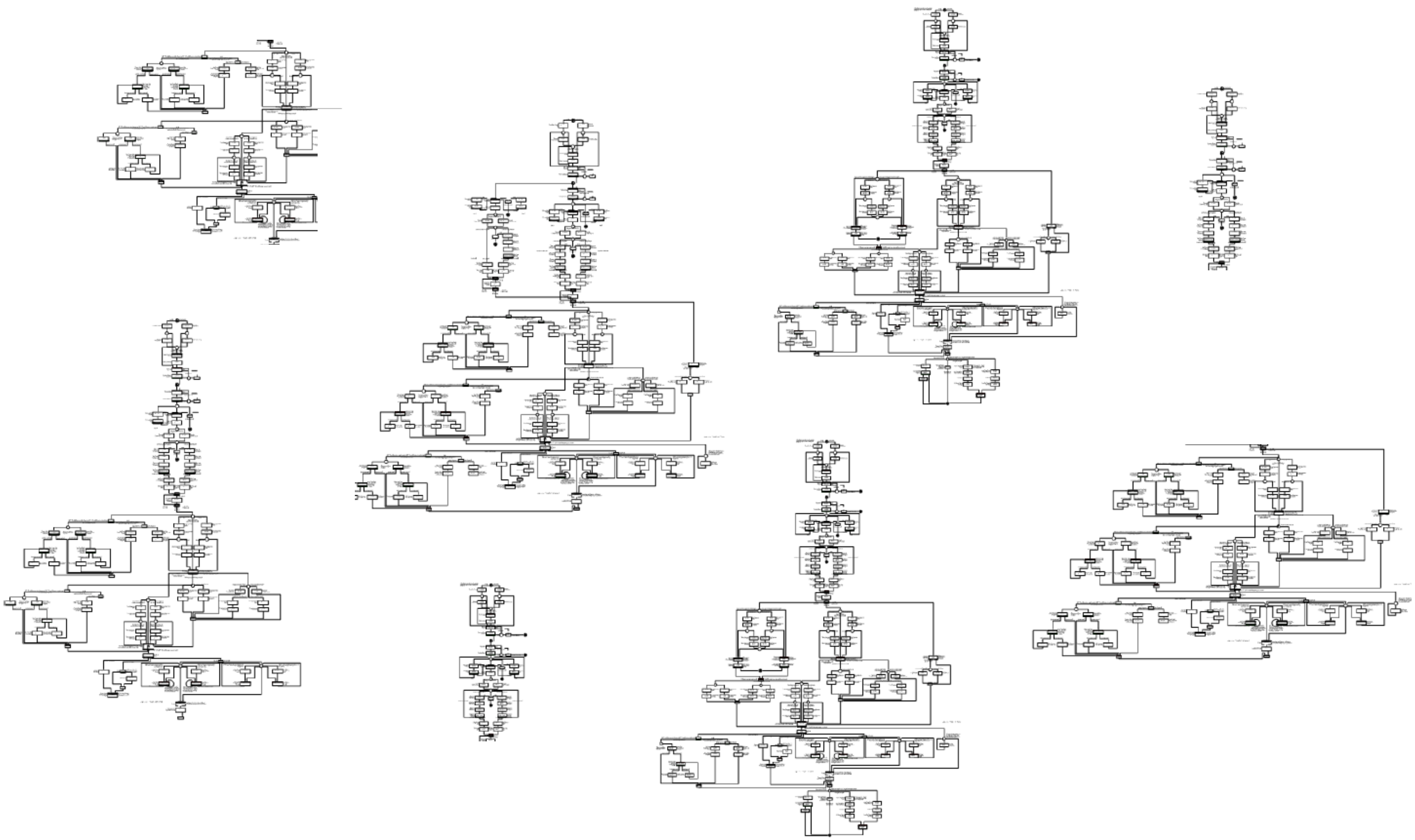
Updating a CPM and related views after a view update!



Agenda

- Challenges & Basic Notions
- Part I: Large Process Models
- Part II: Large Process Model Collections
- Part III: Large Process Structures
- References

Business Process Repositories
Basic Challenge: Dealing with Large Model Collections

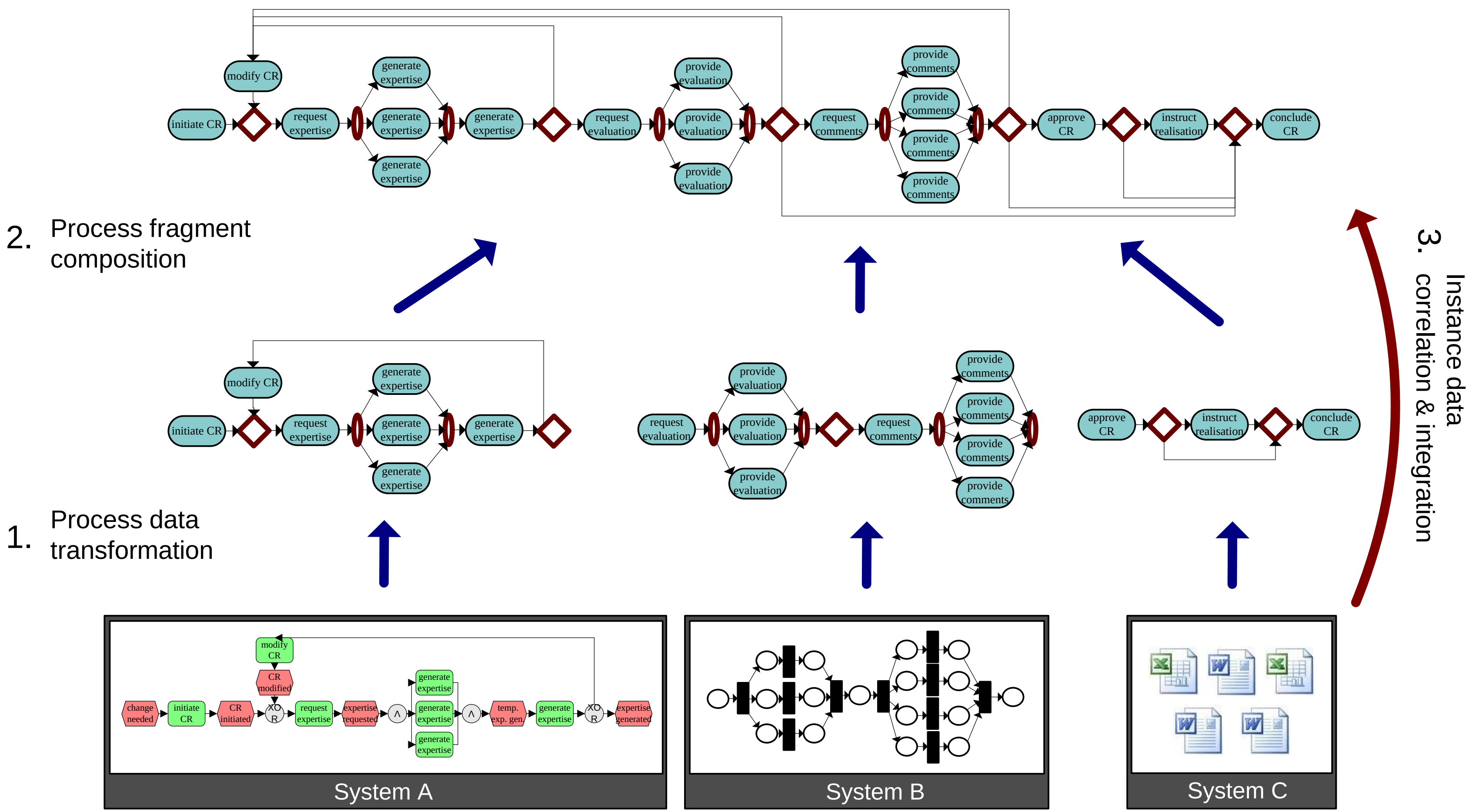


Business Process Repositories

- Standard repository features
  - Check-in / check- out, access control, simple search queries
- Advanced repository features
  - Extract, transform and compose process information
  - Filtering (i.e., clone, detection, similarity search, querying)
  - Managing process variants
  - Merging
  - Navigating in repositories
  - …

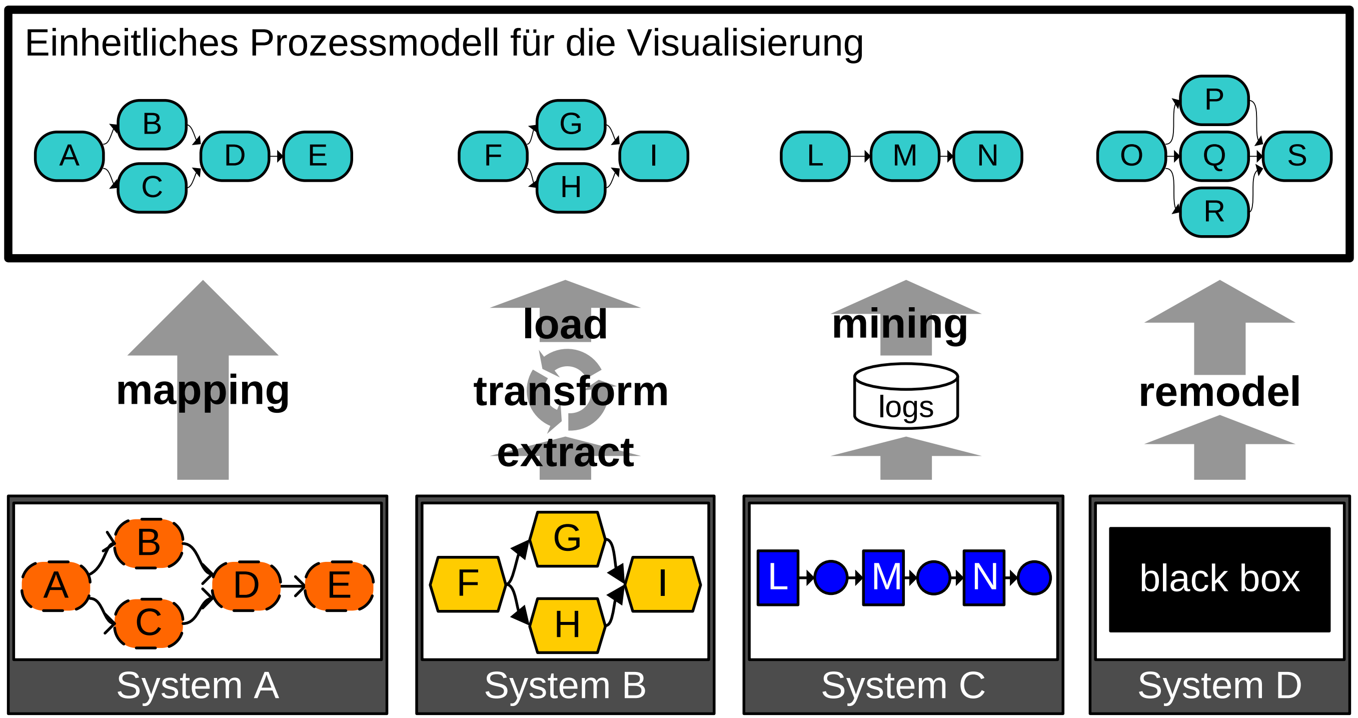
(La Rosa et al. 2011)

Business Process Repositories: Extract, Transform, and Compose Process Fragments



(Bobrik 2008)

Business Process Repositories:
Extract, Transform, and Compose Process Fragments



Business Process Repositories
Filtering

- Filtering
  - Clone detection
    - finding exact matches
  - Similarity search
    - finding close matches
  - Querying
    - looking for patterns

- Searching for similar process models / process fragments
  - Given a query graph, find all process models in the process model collection that are similar

When are two process models similar?

(Dijkman et al. 2011)

When are two process models similar?

- Similarity measure defines when two process models are similar
  - Value between 0 and 1
  - 0 indicates no similarity and 1 indicates identical elements

(Dijkman et al. 2011)

When are two process models similar?

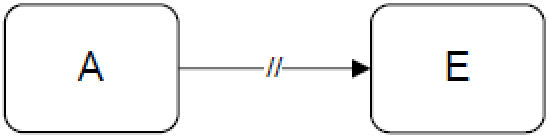
- Label matching similarity
  - Similarity based on the labels of business process model elements
- Structural similarity
  - Measures similarity based on the labels of a business process model element, as well as the relations between these elements
- Behavior similarity
  - Measure similarity based on the intended behavior of process models

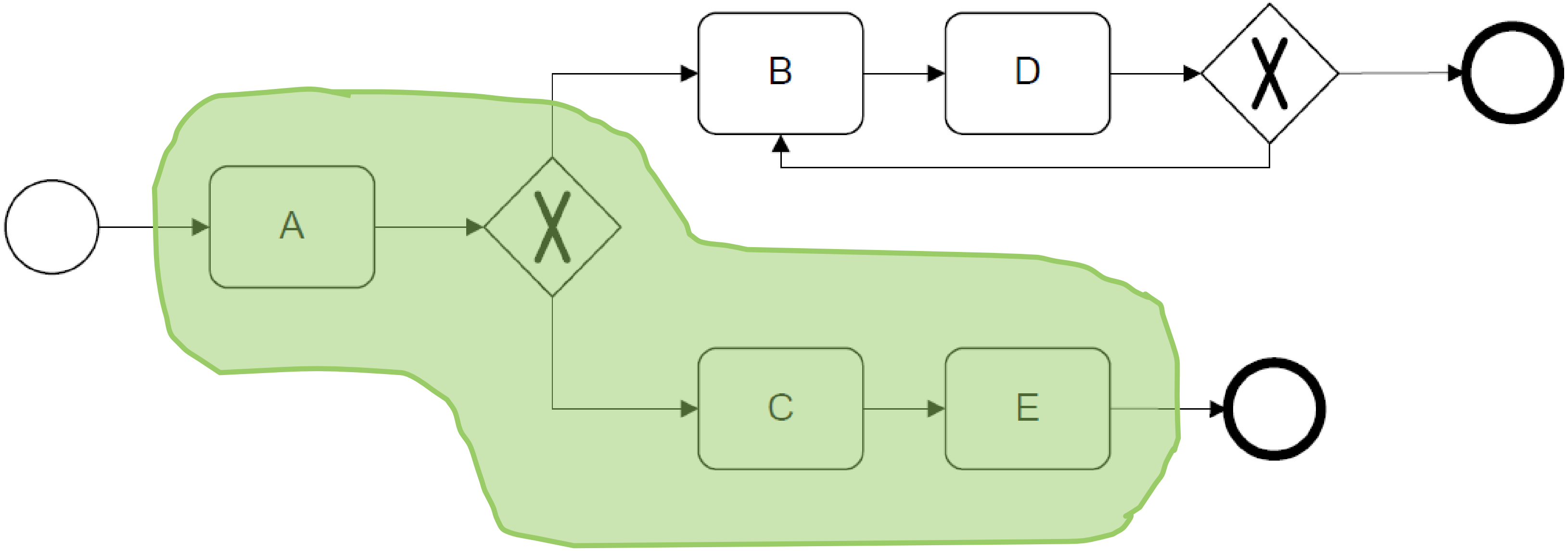
(Dijkman et al. 2011)

Business Process Repositories
Filtering: Querying Process Models

- Querying a process model for patterns
  - Given a query graph, find all matching sub-graphs in the process model collection
  - Example of such a query language
    - BPMN-Q (Awad 2007): Queries expressed in BPMN; usage of wildcard nodes and arcs possible
  - Efficient querying of process fragments
    - Index for efficient querying (Yan, Dijkman, and Grefen 2012)

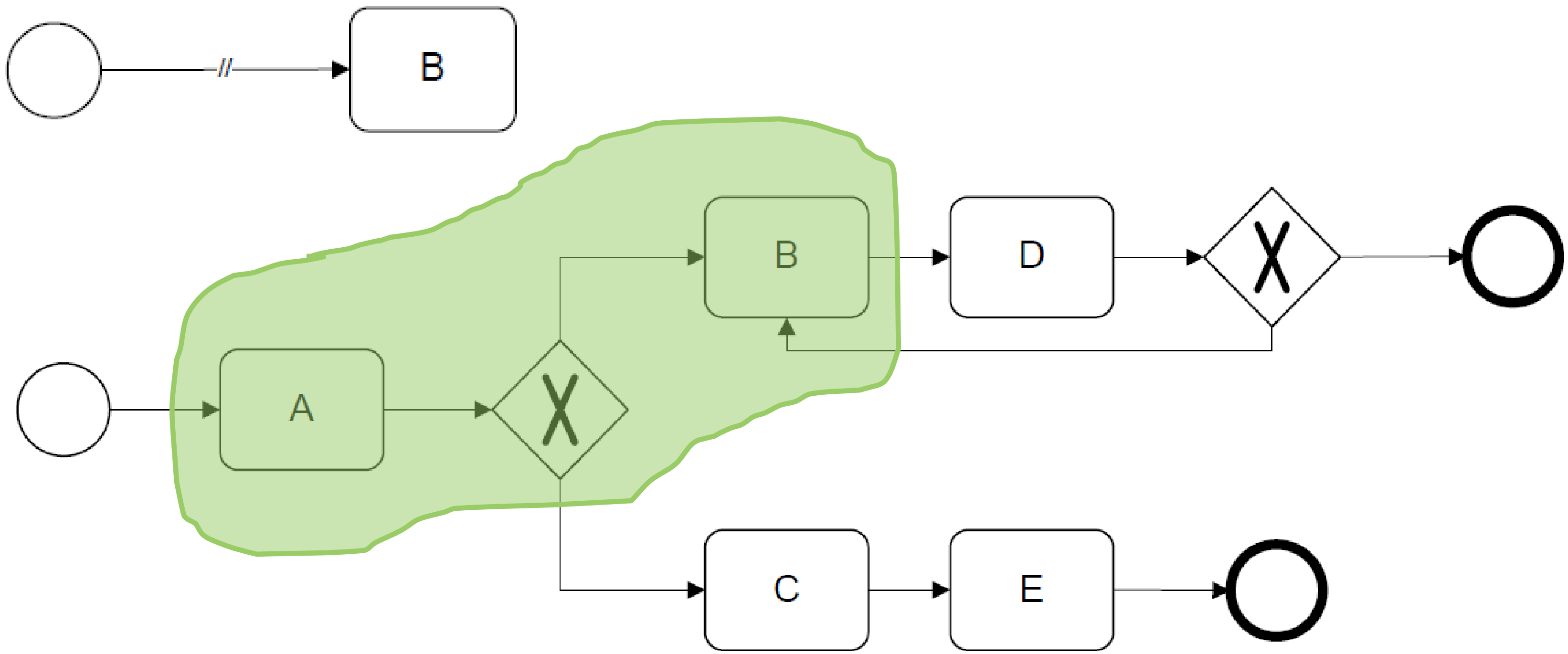
- Looking for patterns in a process model
  - What happens between A and E?





(Awad 2007, Awad et al. 2008)

- Looking for patterns in a process model
  - What happens from the start until B is reached?



(Awad 2007, Awad et al. 2008)

Agenda

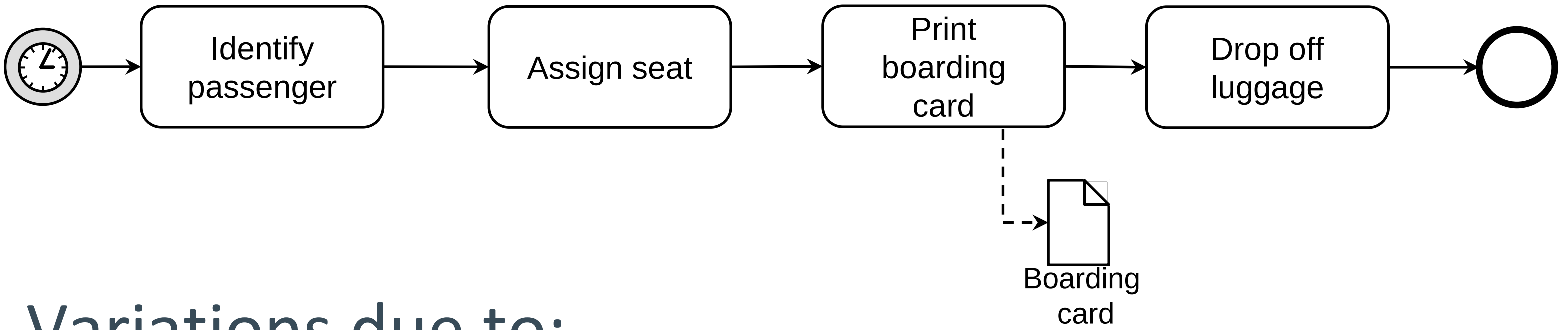
- Challenges & Basic Notions
- Part I: Large Process Models
- Part II: Large Process Model Collections
- Part III: Large Process Structures
- References

Variability

Sources of Variations

Industrial Example
Flight Check-in

- Generic process:

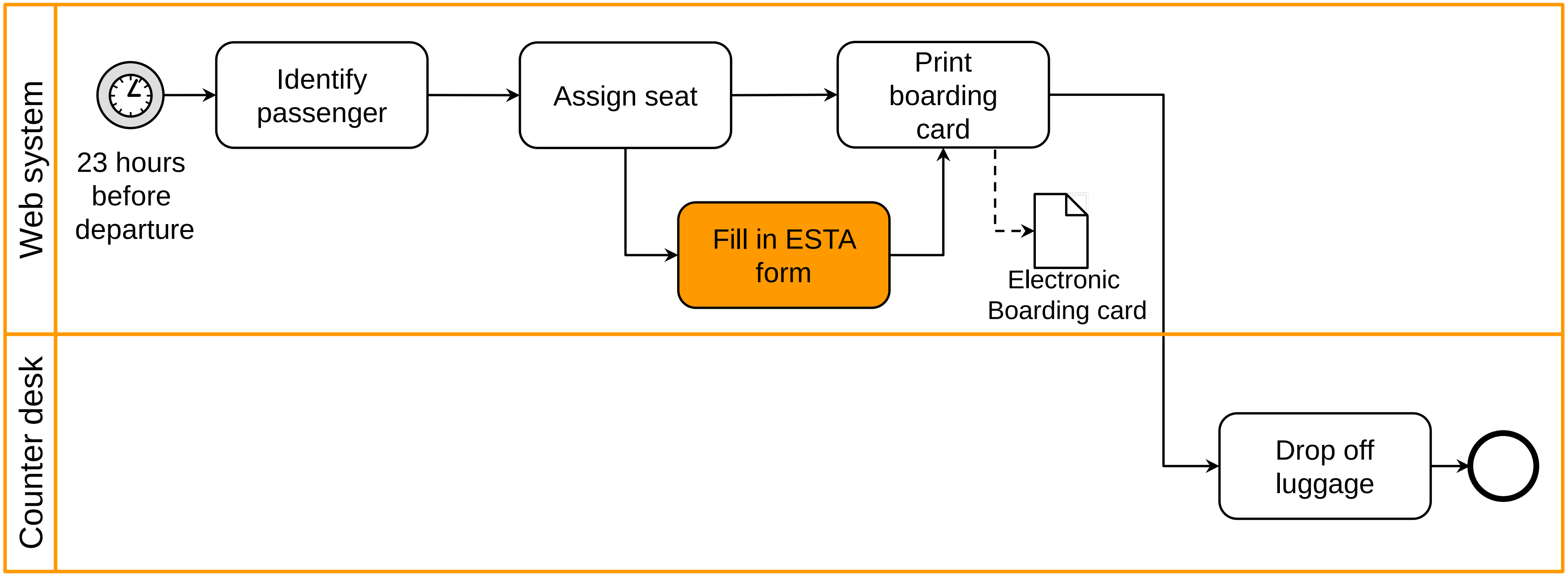


- Variations due to:
  - Airline policies (e.g., check-in opening time)
  - Type of passenger (e.g., adult, child, handicapped)
  - Type of ticket (e.g., economy, business)
  - Type of carried items (e.g., pet)
  - etc.

» **Hundreds of variants**

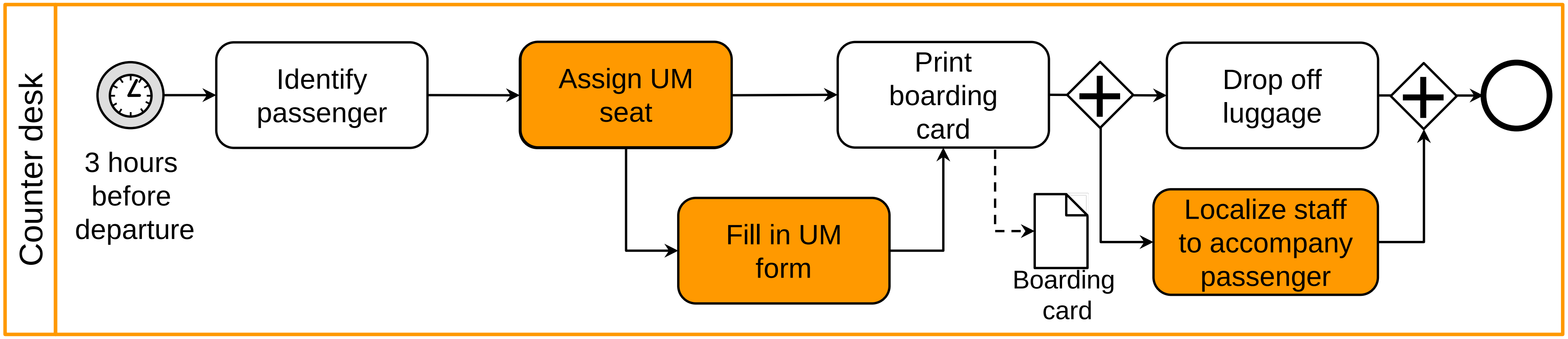
Industrial Example
Flight Check-in

- Variant 1:
  - Online check-in
  - Adult flying with a business class ticket
  - International flight from EU to USA

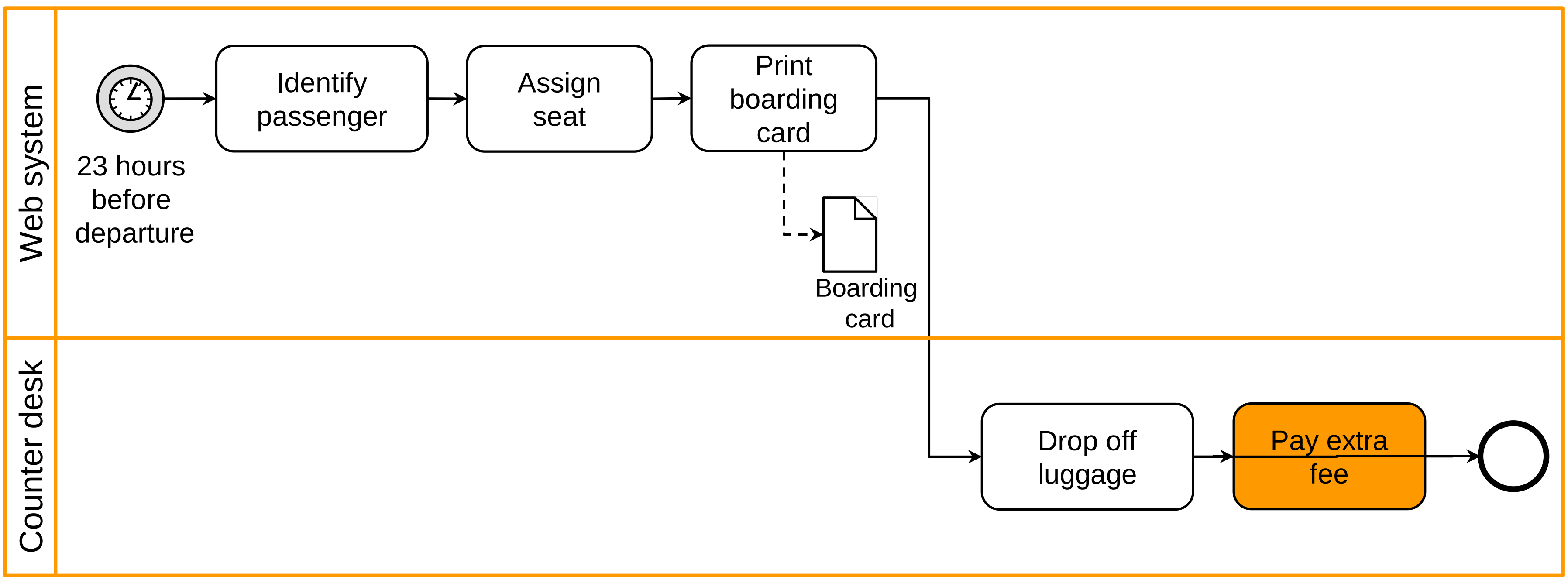


Industrial Example
Flight Check-in

- Variant 2:
  - Counter desk check-in
  - Unaccompanied Minor (UM)



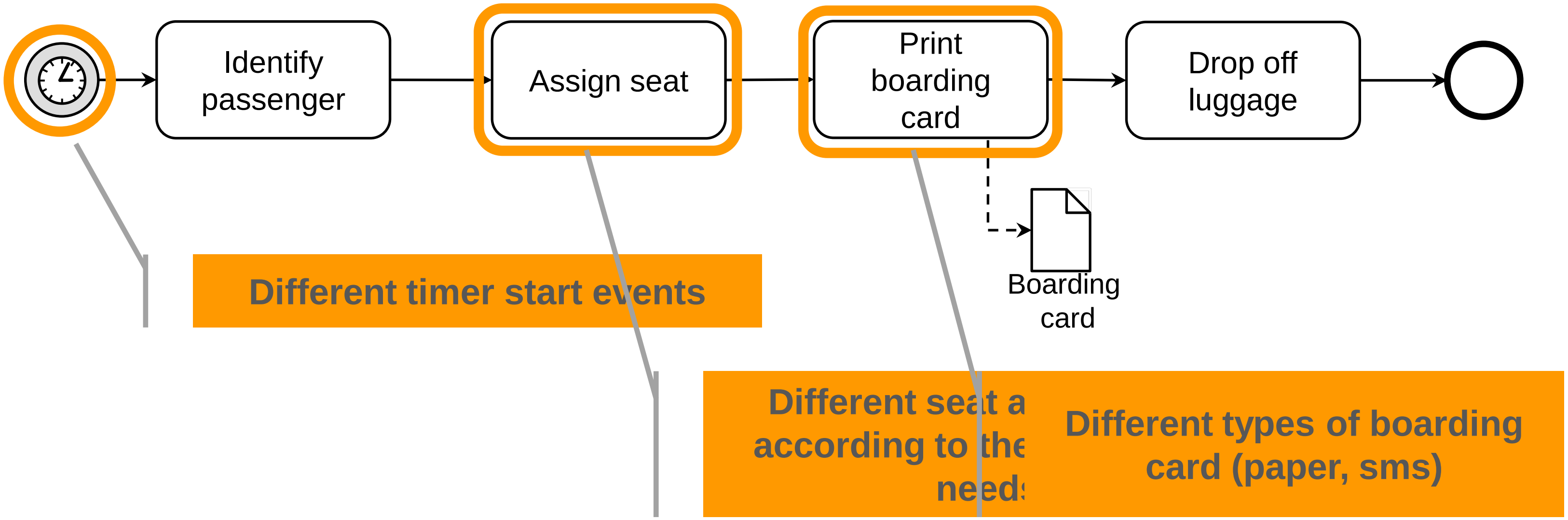
- Variant 3:
  - Online check-in
  - Adult carrying overweight



How are Business Processes which are subject to variability characterized?

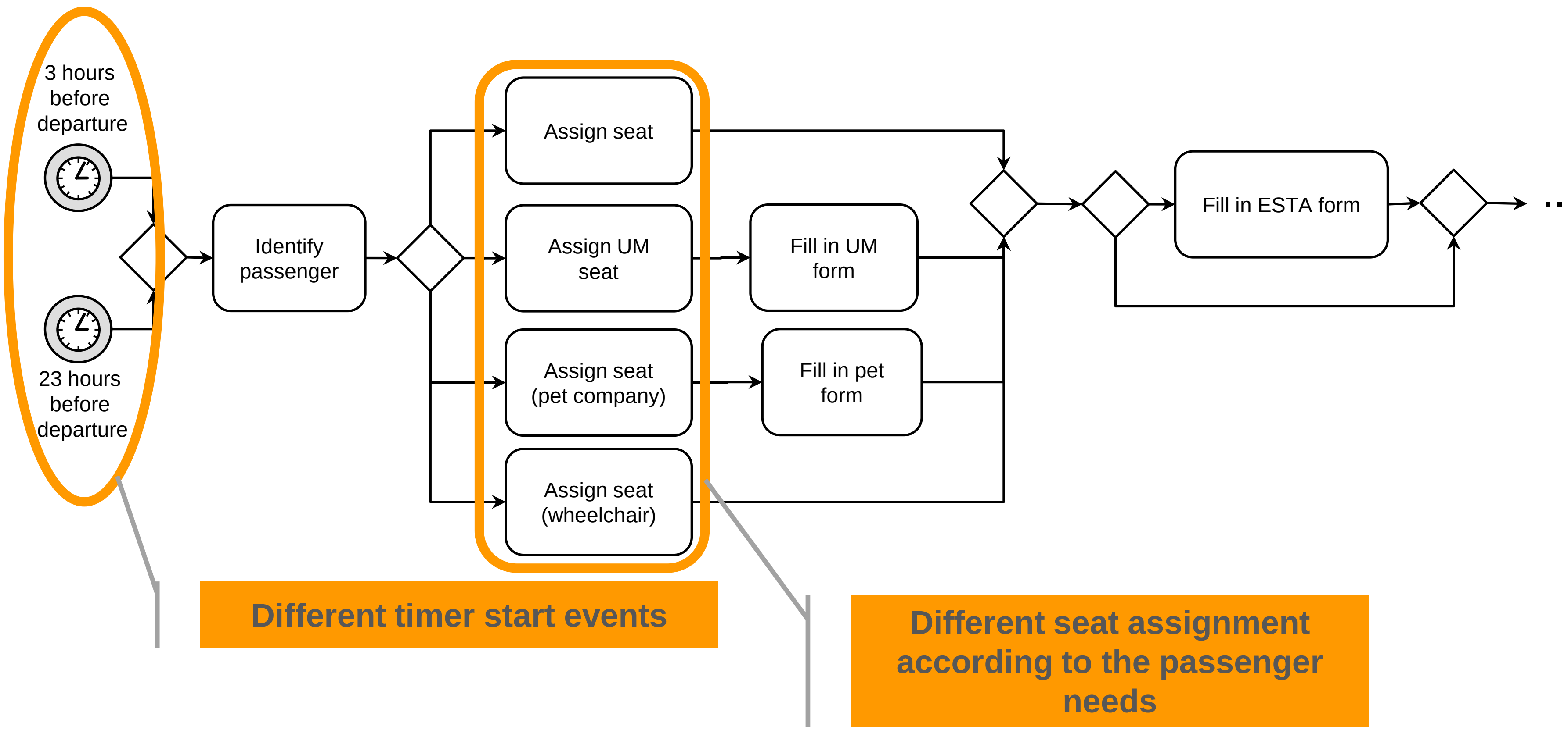
Representing Variability Concepts

- Variable parts
  - Parts being subject to variation (commonly known as variation point)



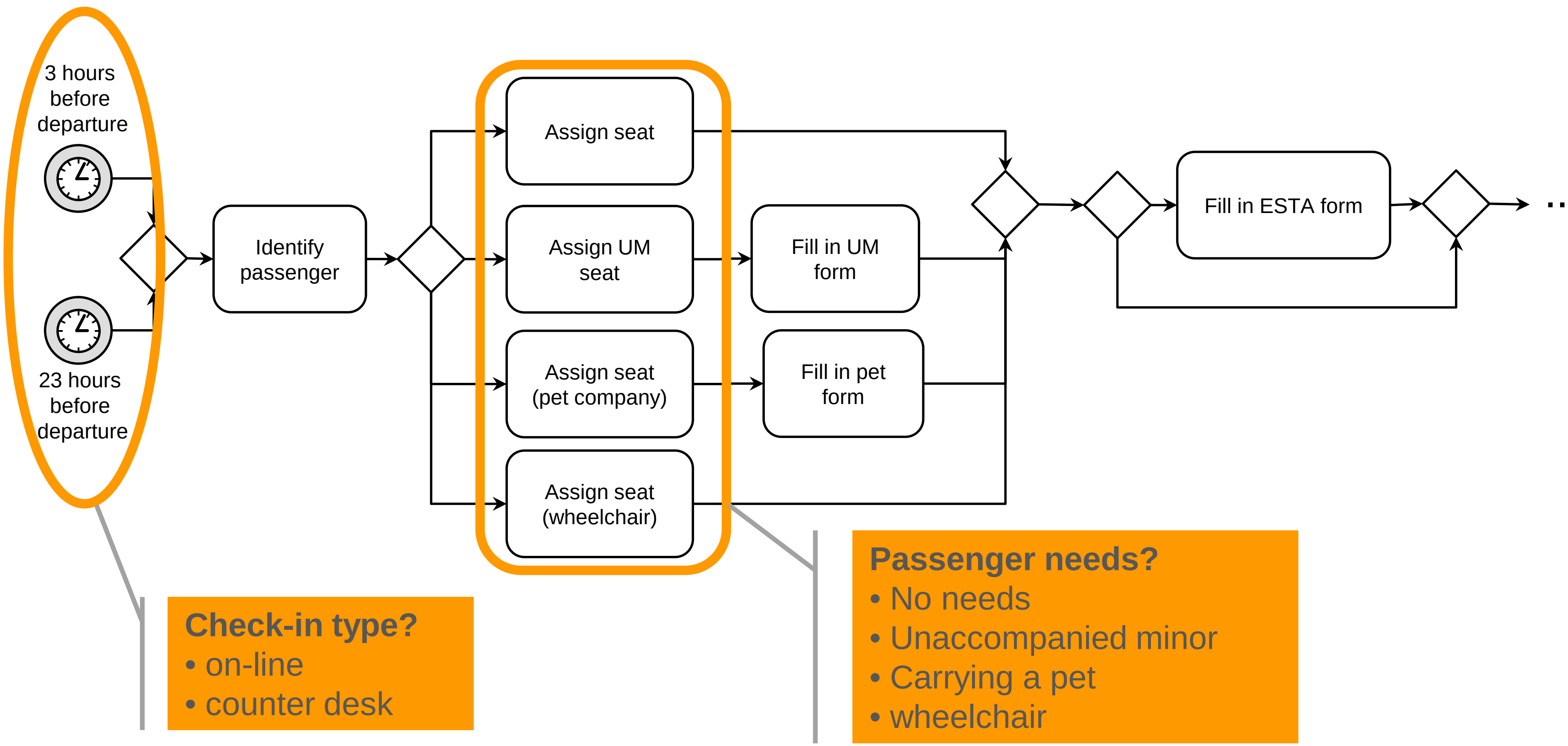
Representing Variability Concepts

- Alternatives that exist for each variable parts
  - Alternatives that fit in each variable part
  - Relationship constraints between such alternatives

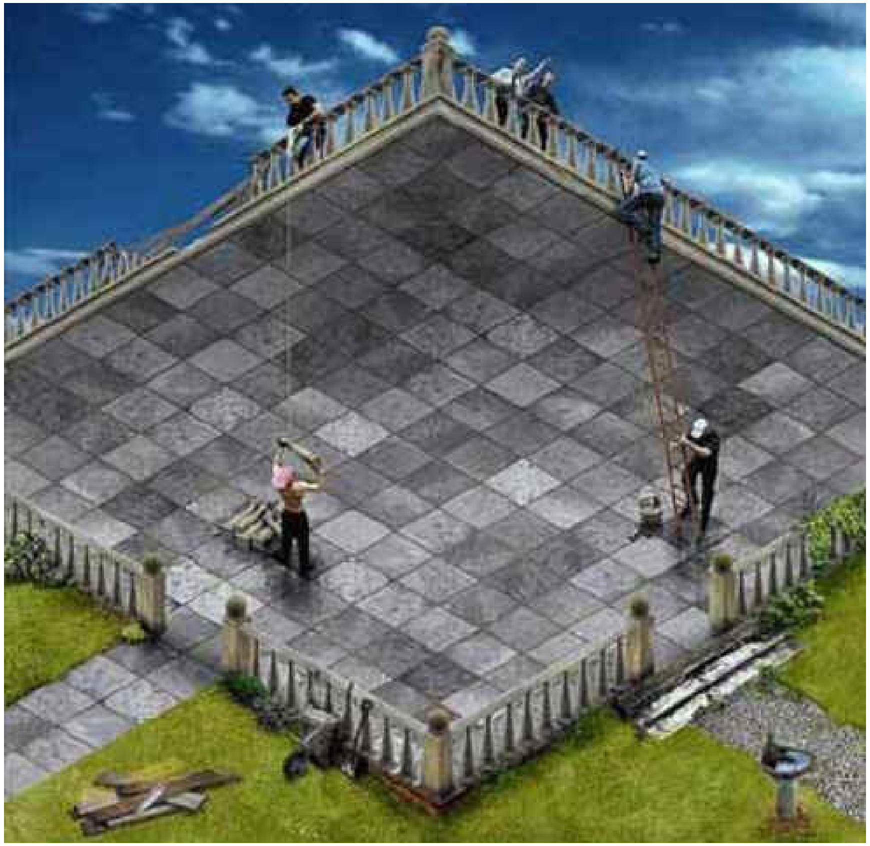


Representing Variability Concepts

- Application context for each alternative
  - Conditions that make these variables being applied
  - Usually represented by a set of variables

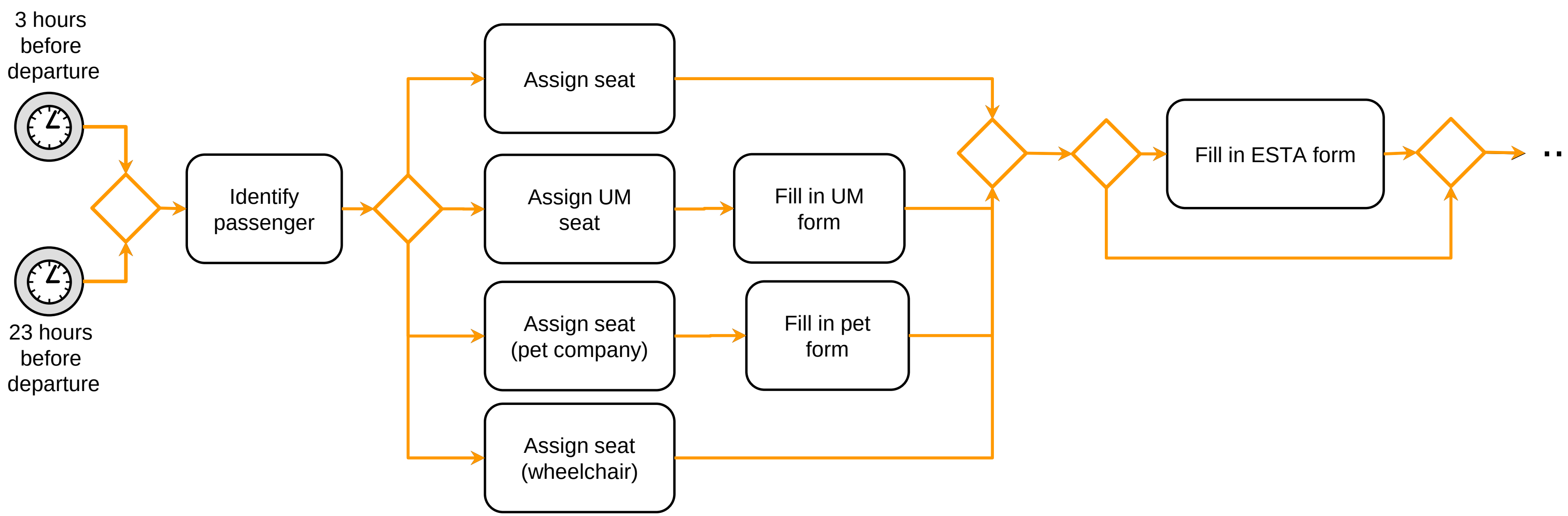


- Where can we find variability in a BP?
  - Behavioural
  - Functional
  - Informational
  - Organizational
  - Temporal
  - Operational



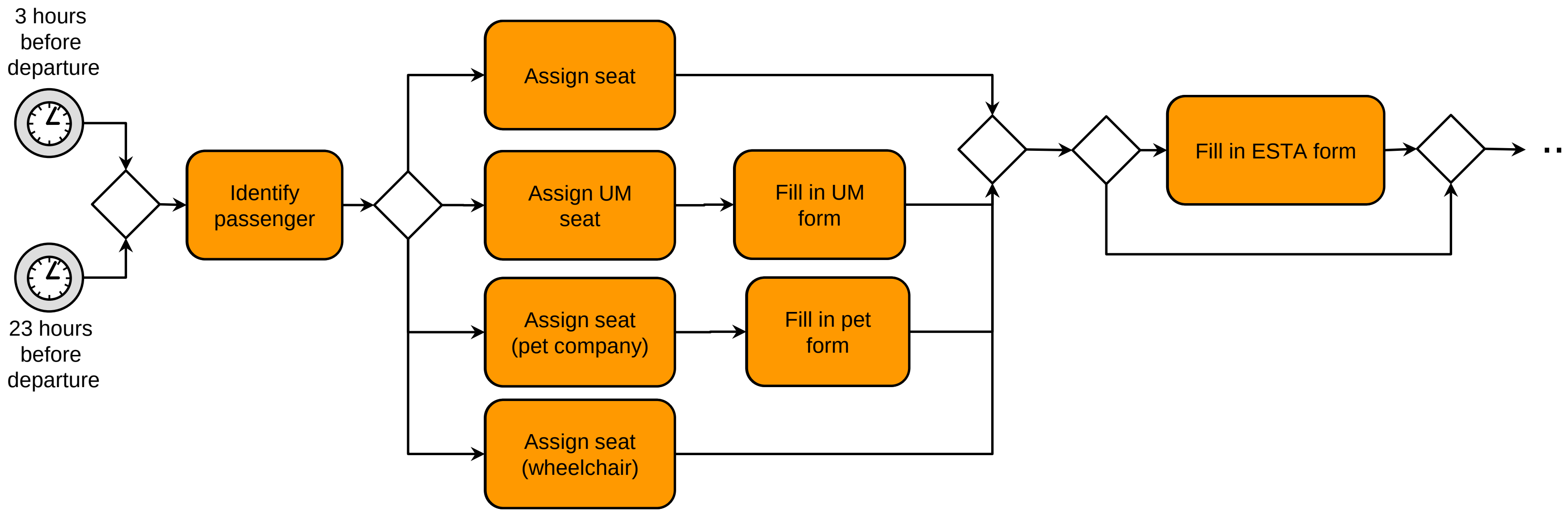
- Behavioural

It captures the dynamic behavior of a BP model and corresponds to the control flow between the activities. A control flow schema includes information about the order of the activities or the constraints for their execution.



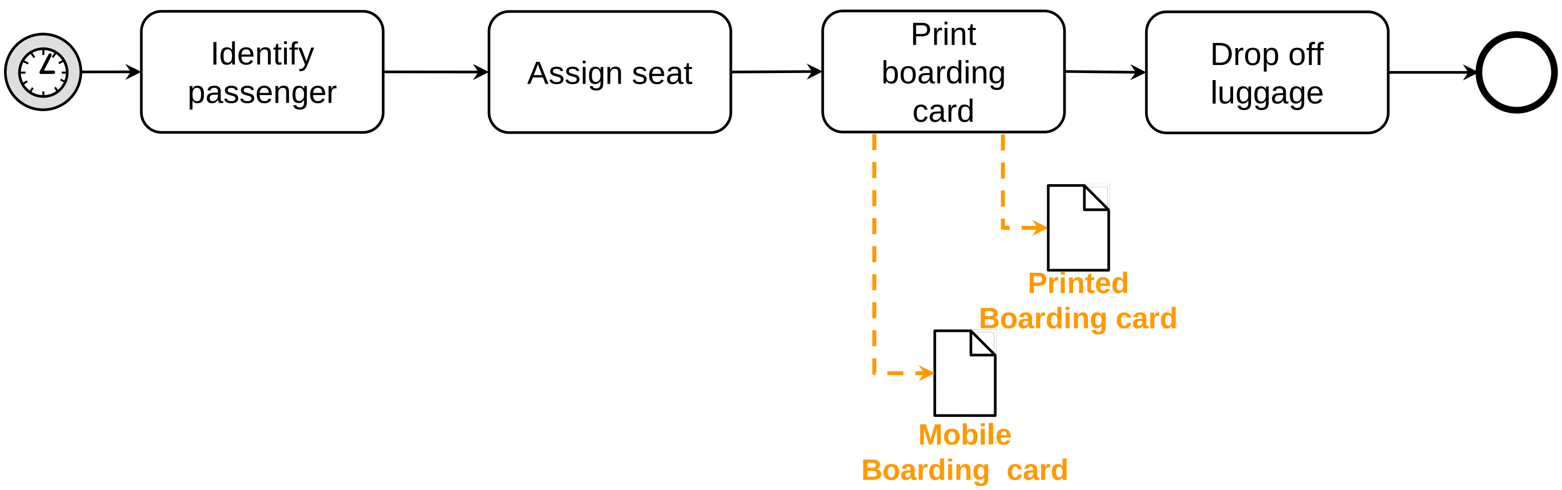
- Functional

> It specifies the decomposition of BPs, i.e., it represents the activities to be performed. While an atomic activity is associated with a single action, a complex activity refers to a subprocess or, more precisely, a sub-process model.



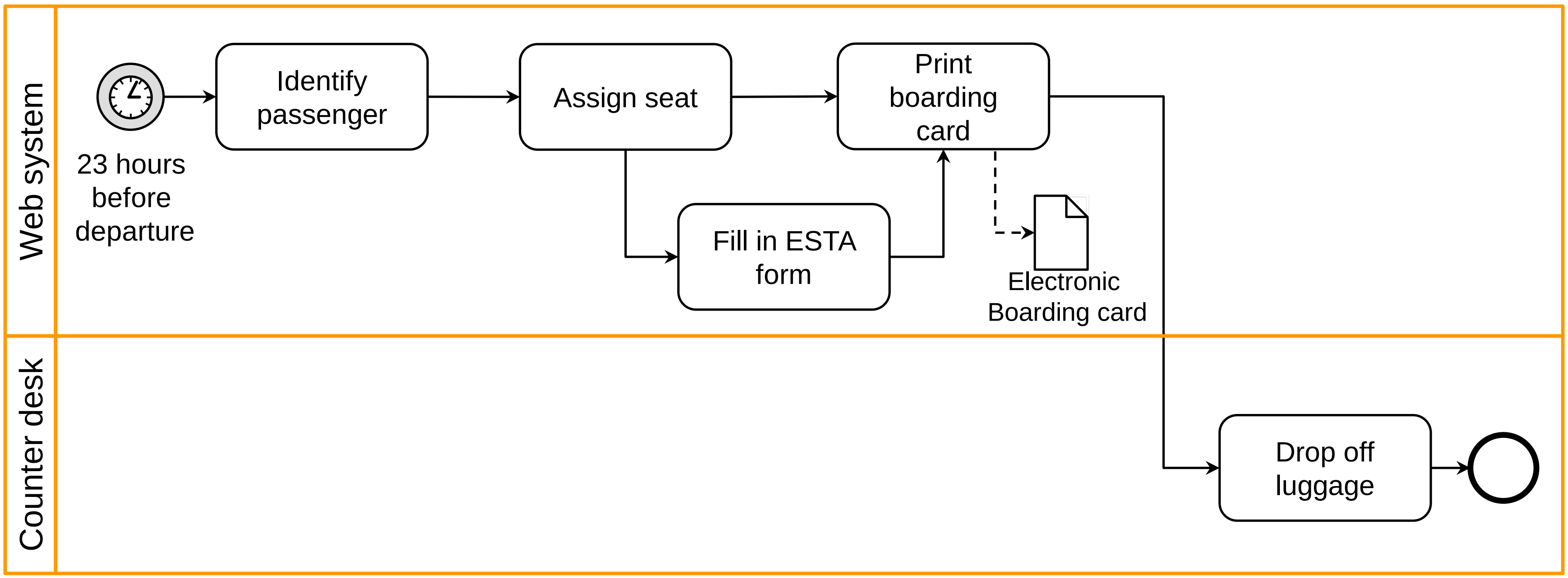
- Informational

It concerns data and data flow, i.e., it represents the informational entities (e.g., data, artifacts, products, and objects) consumed or produced during the execution of BP activities.



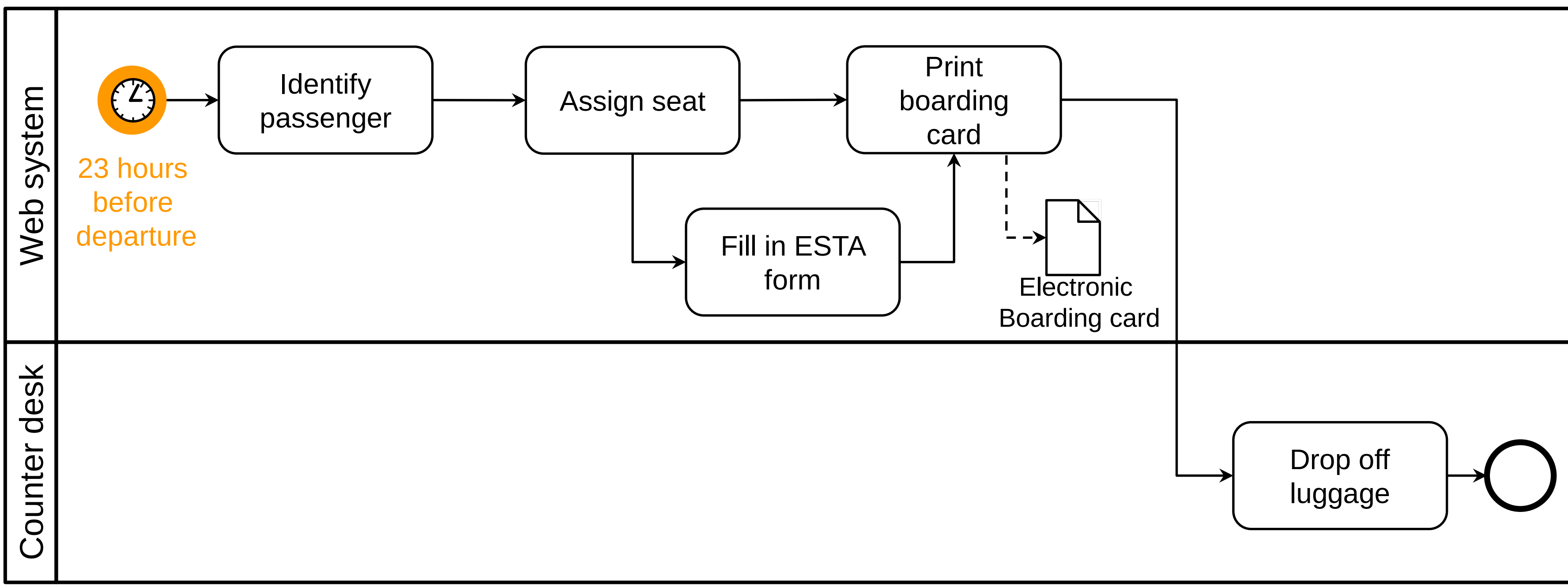
- Organizational

It deals with the assignment of resources to the activities of a BP model, i.e., it represents the actors, roles (i.e., humans or systems), within an organization being in charge of executing certain BP activities.



- Temporal

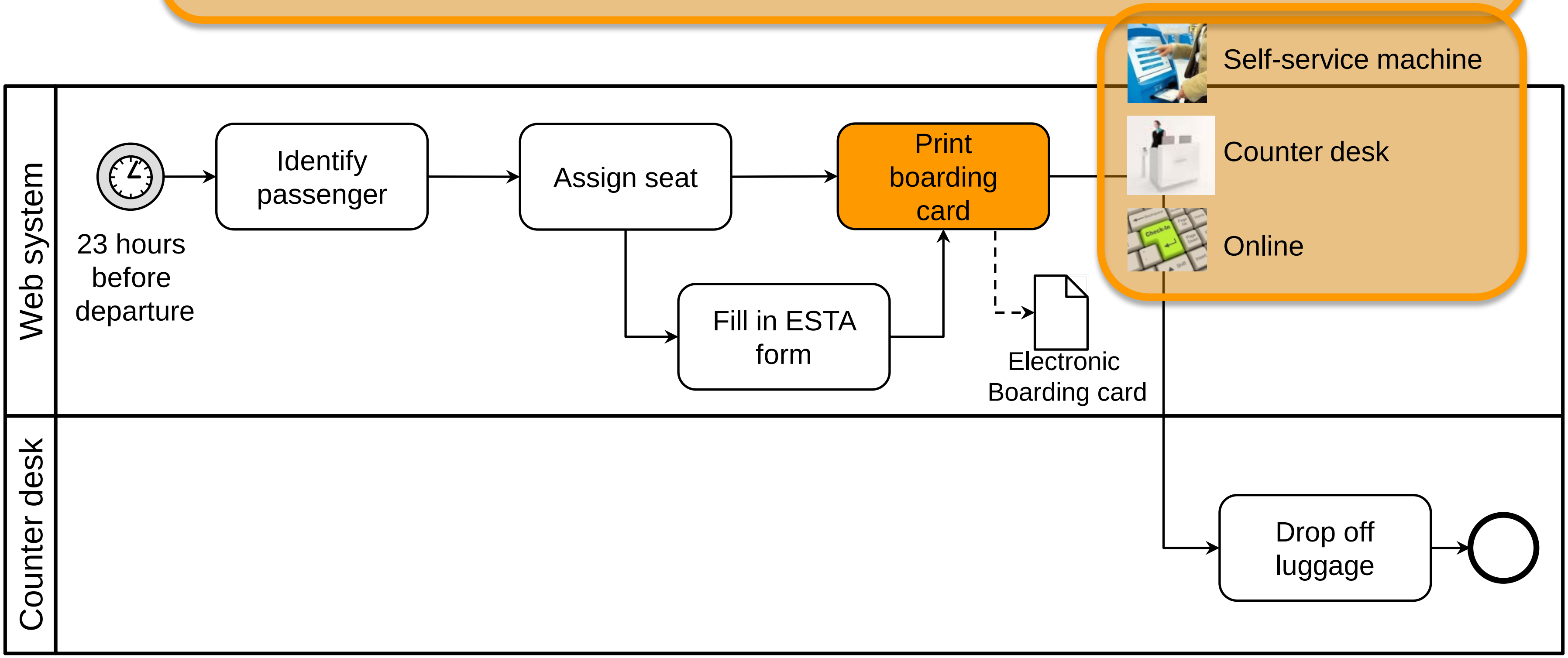
It deals with time issues and temporal constraints, i.e., it represents the occurrence of events during the course of a process, which affects the scheduling of activities from this process.



Process Perspectives

- Operational

It refers to the implementation of process activities, i.e., the application services to be executed when an atomic activity is performed.



Limitations of Common BPMLs

Are common BPML constructs sufficient to properly represent variability in BPs?

Limitations of Common BPMLs

Could we use...

- Conditional branching,
- Separate process models, or
- Sub-processes

... for such purpose?

Limitations of Common BPMLs

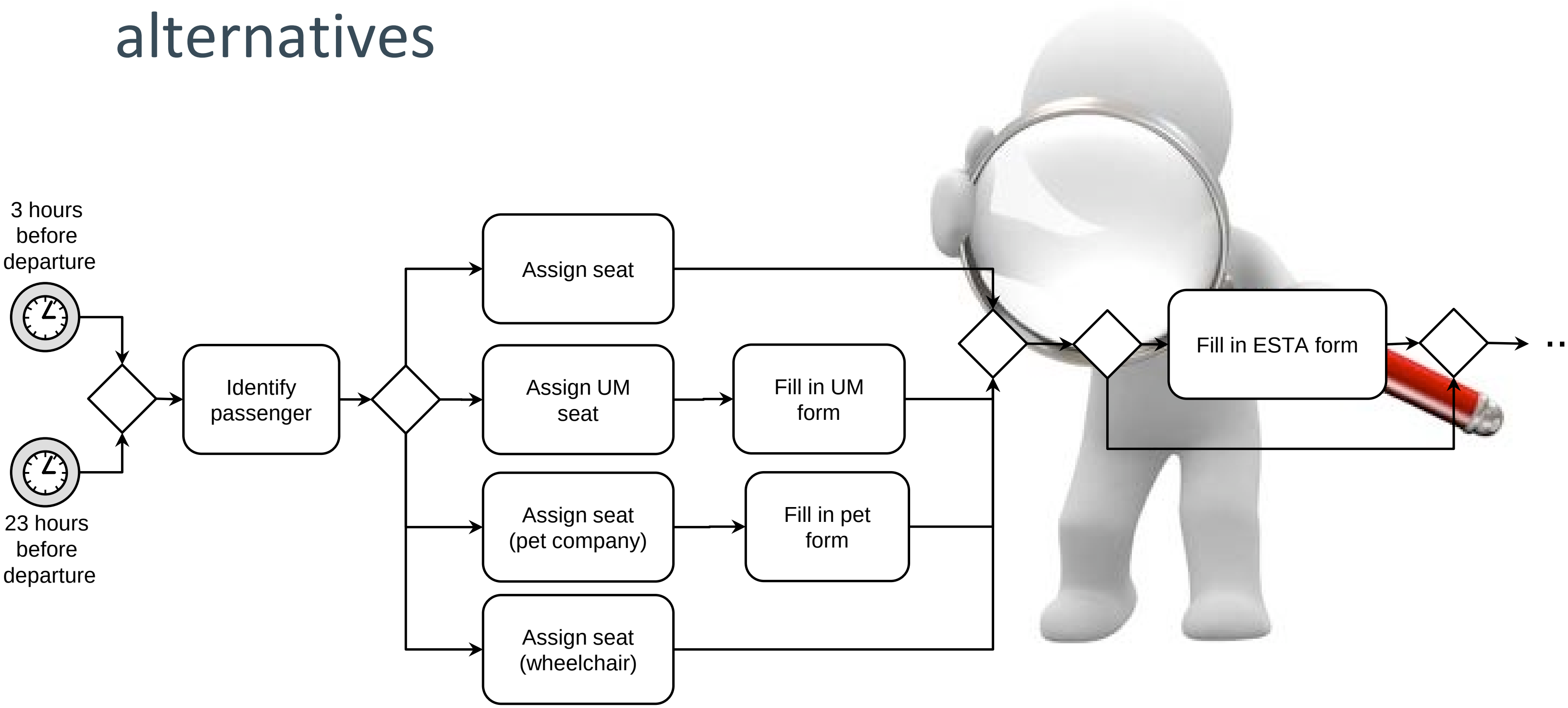
Could we use...

- Conditional branching,
- Separate process models, or
- Sub-processes

... for such purpose?

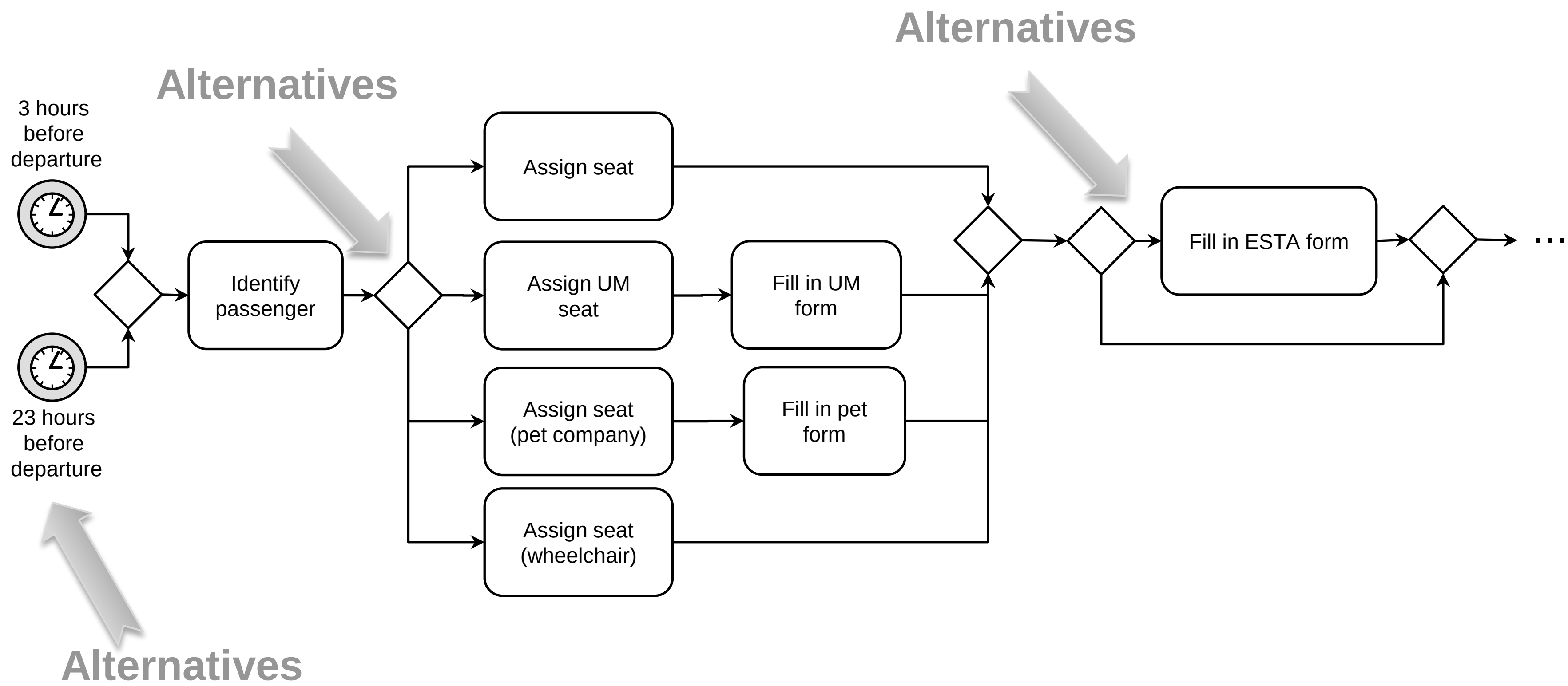
Limitations of Common BPMLs
Conditional Branching

- BPMLs could be used to represent process variant alternatives



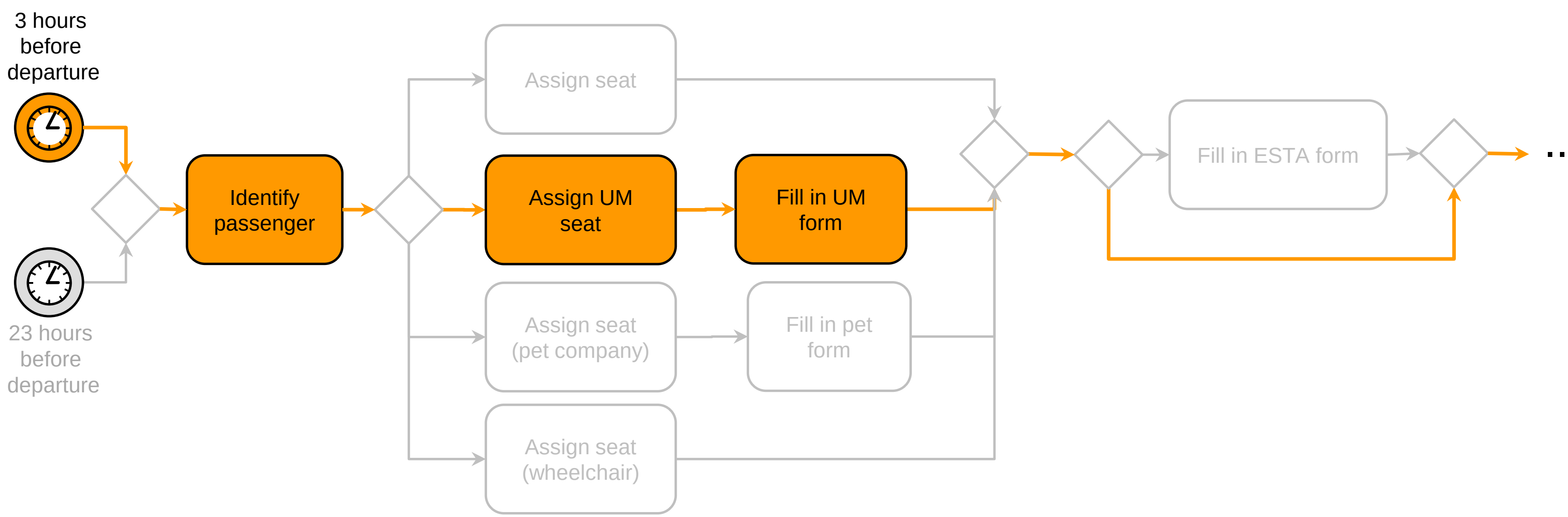
Limitations of Common BPMLs
Conditional Branching

- (+) Allows visualizing *all* alternatives in *one shoot*



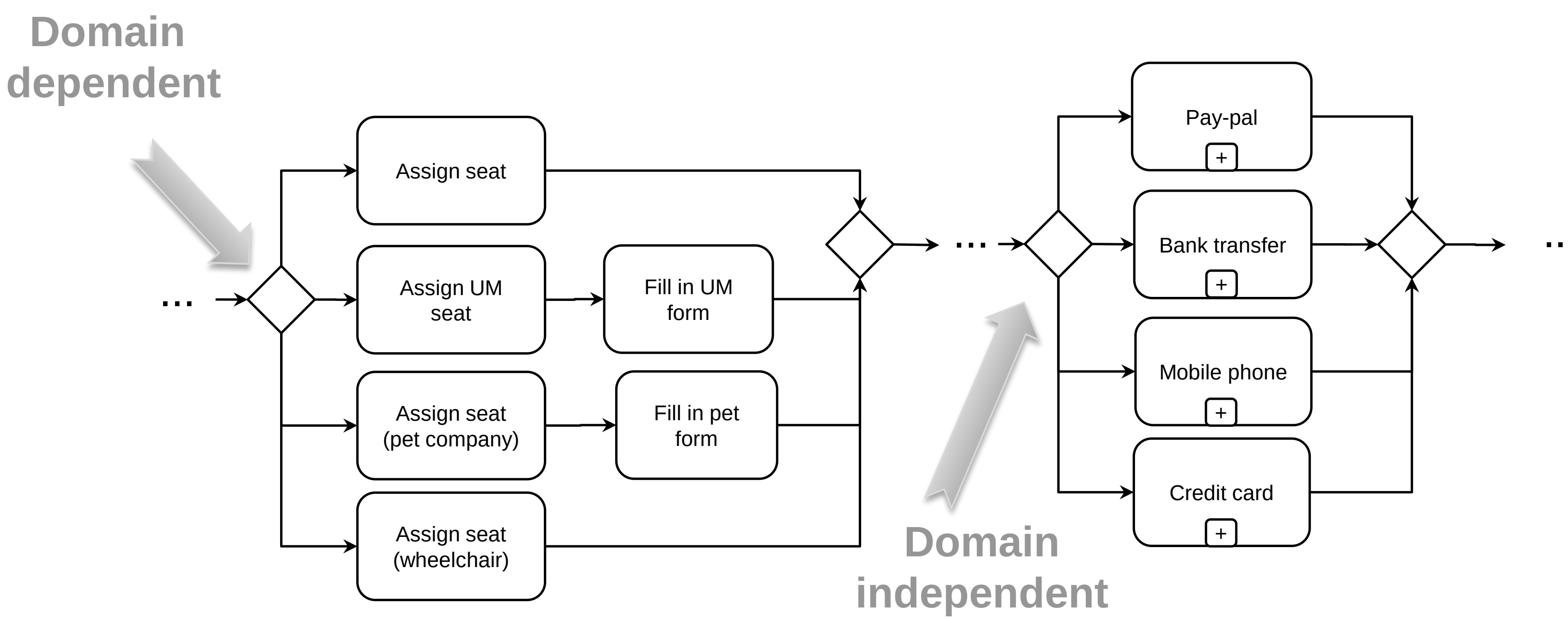
Limitations of Common BPMLs
Conditional Branching

- (-) Hinders identifying existing variants



Limitations of Common BPMLs
Conditional Branching

- (-) Hinders differentiating between *domain dependent* and *domain independent* conditional branching



Limitations of Common BPMLs

Could we use...

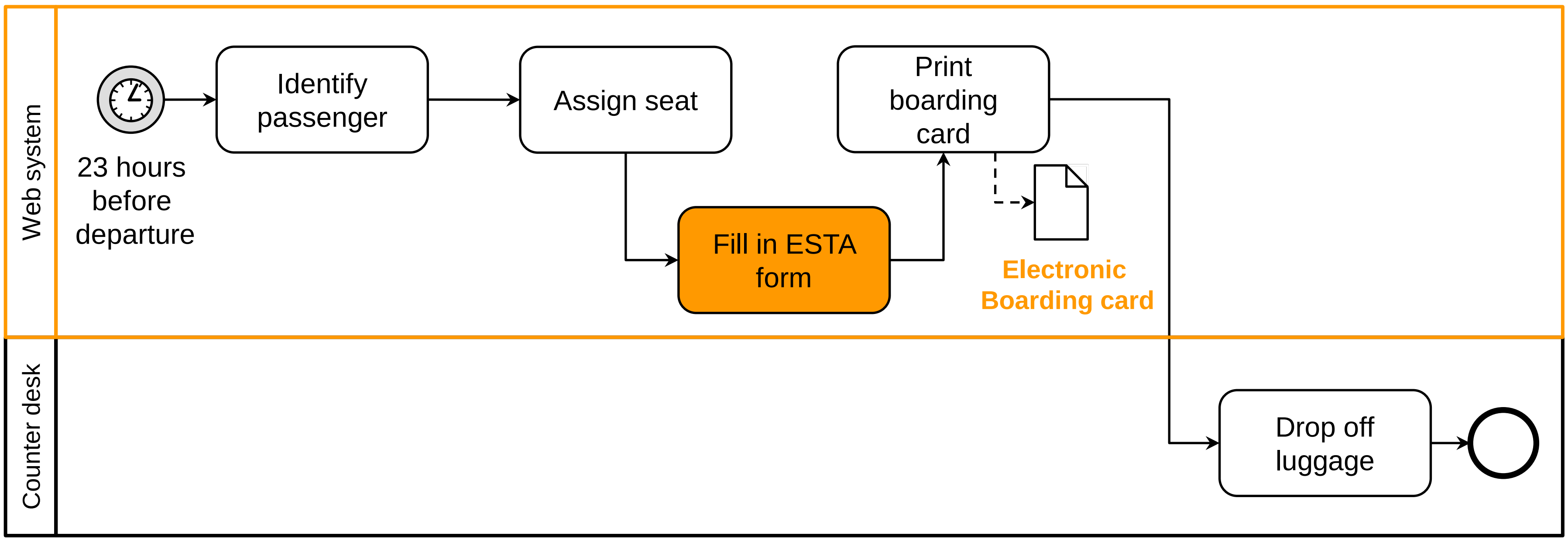
– Conditional branching,
– Separate process models, or
– Sub-processes

... for such purpose?

Limitations of Common BPMLs
Separate process models

- (+) Each process variant in a separate model

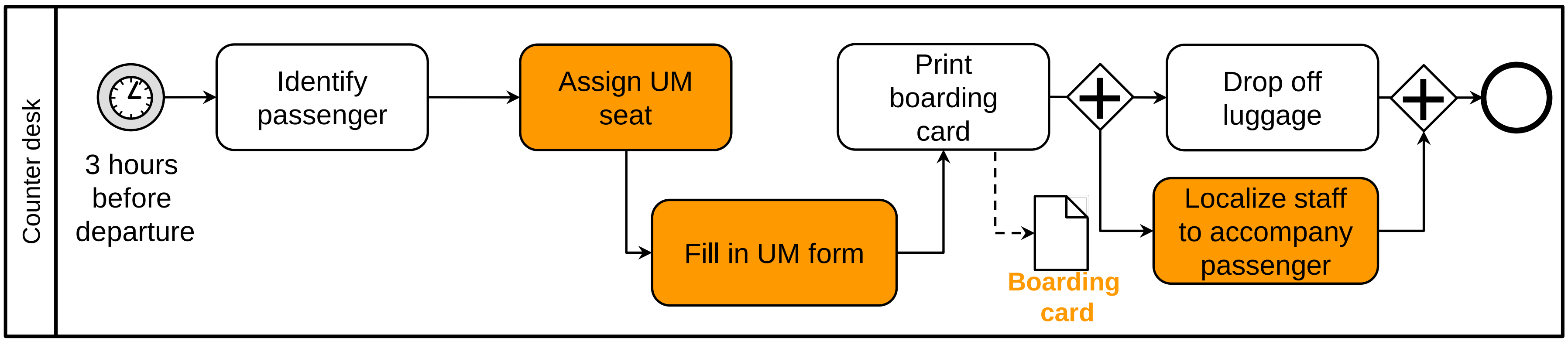
Variant 1:



Limitations of Common BPMLs
Separate process models

- (+) Each process variant in a separate model

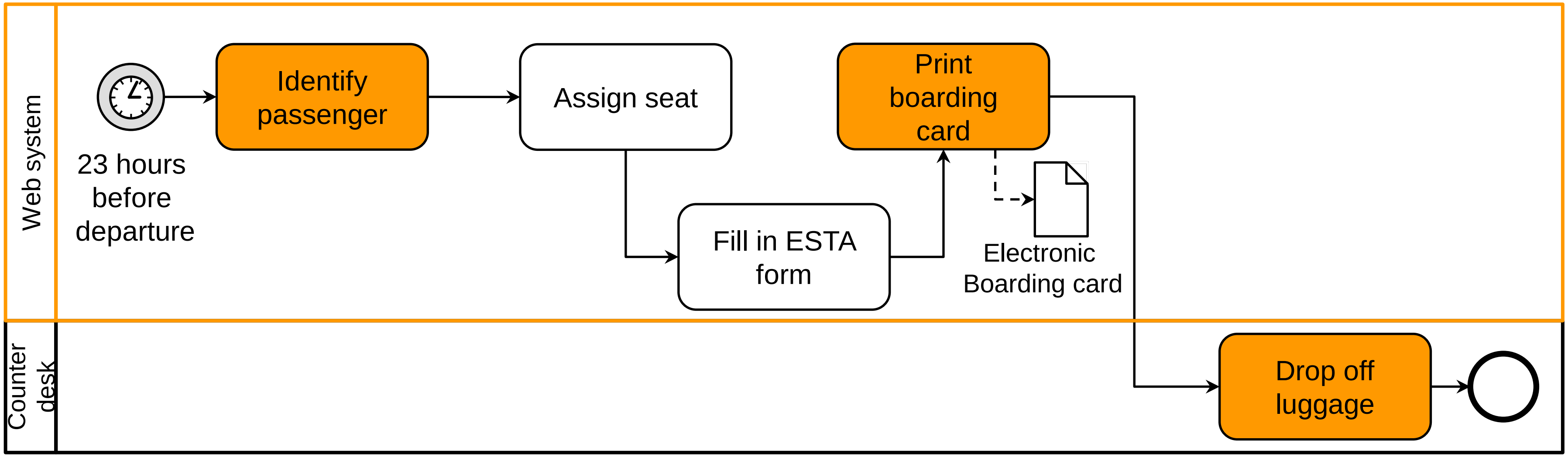
Variant 2:



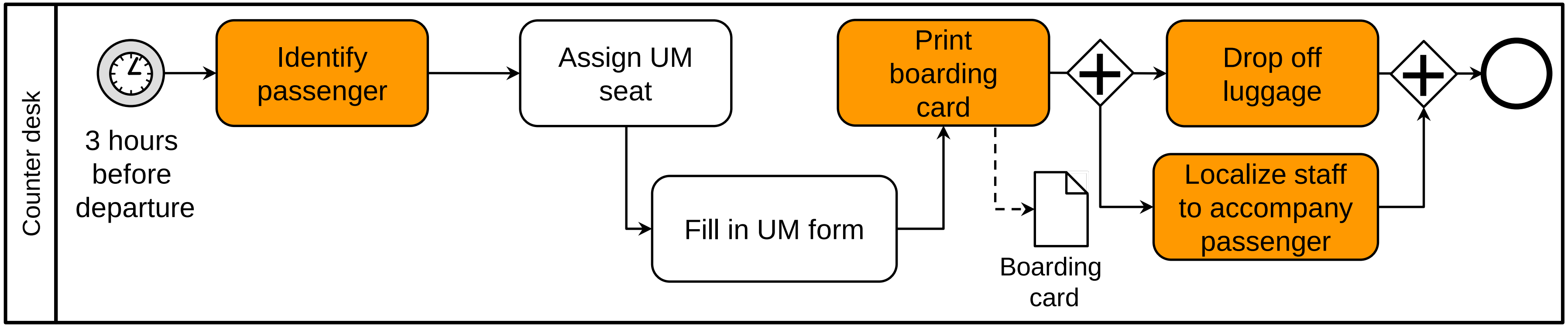
Limitations of Common BPMLs
Separate process models

- (-) Common fragments have to be replicated

Variant 1:



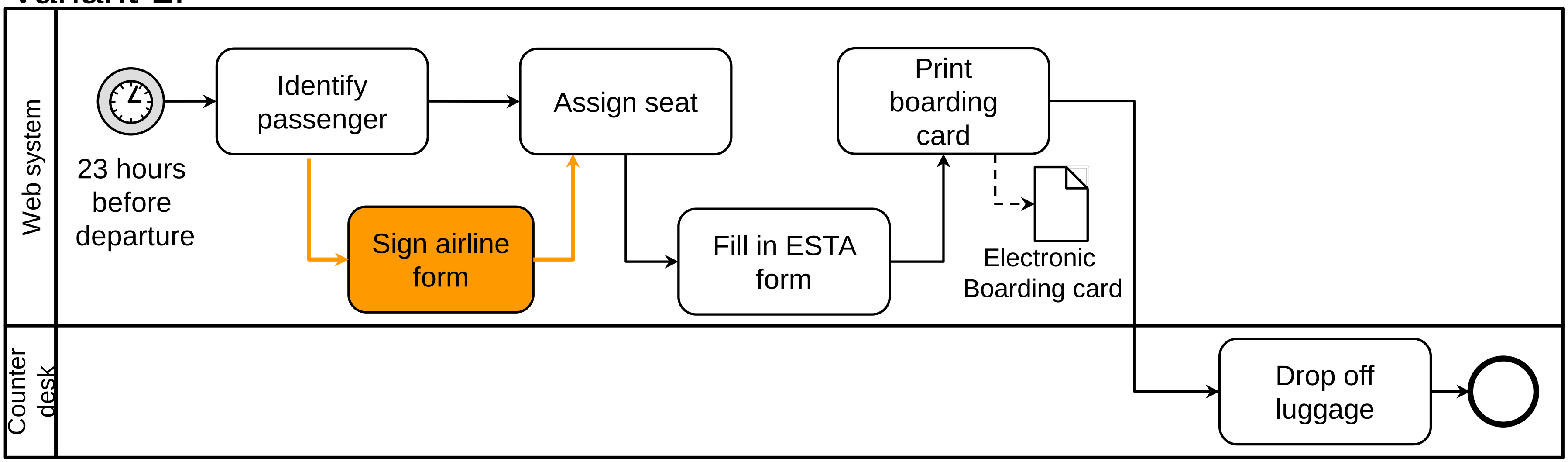
Variant 2:



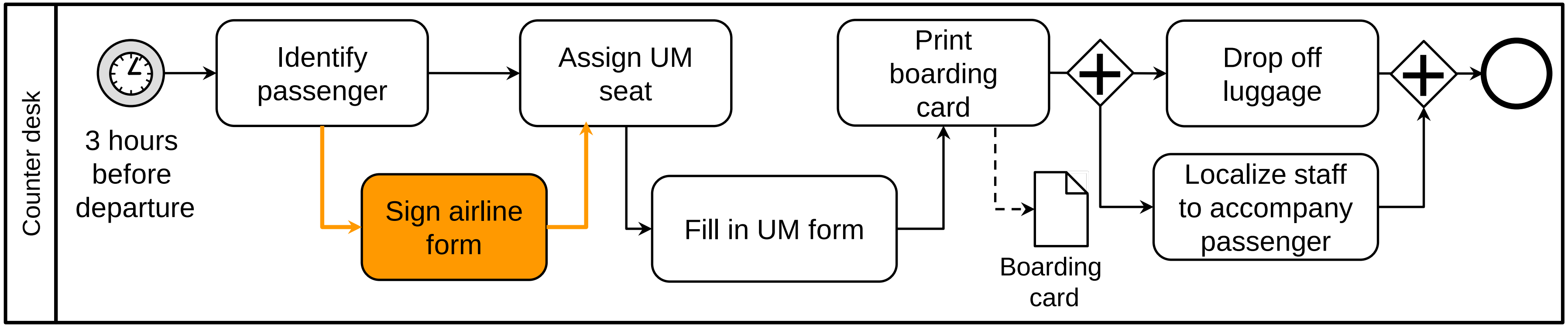
Limitations of Common BPMLs
Separate process models

- (-) Changes need to be replicated in all variants

Variant 1:



Variant 2:



Limitations of Common BPMLs

Could we use...

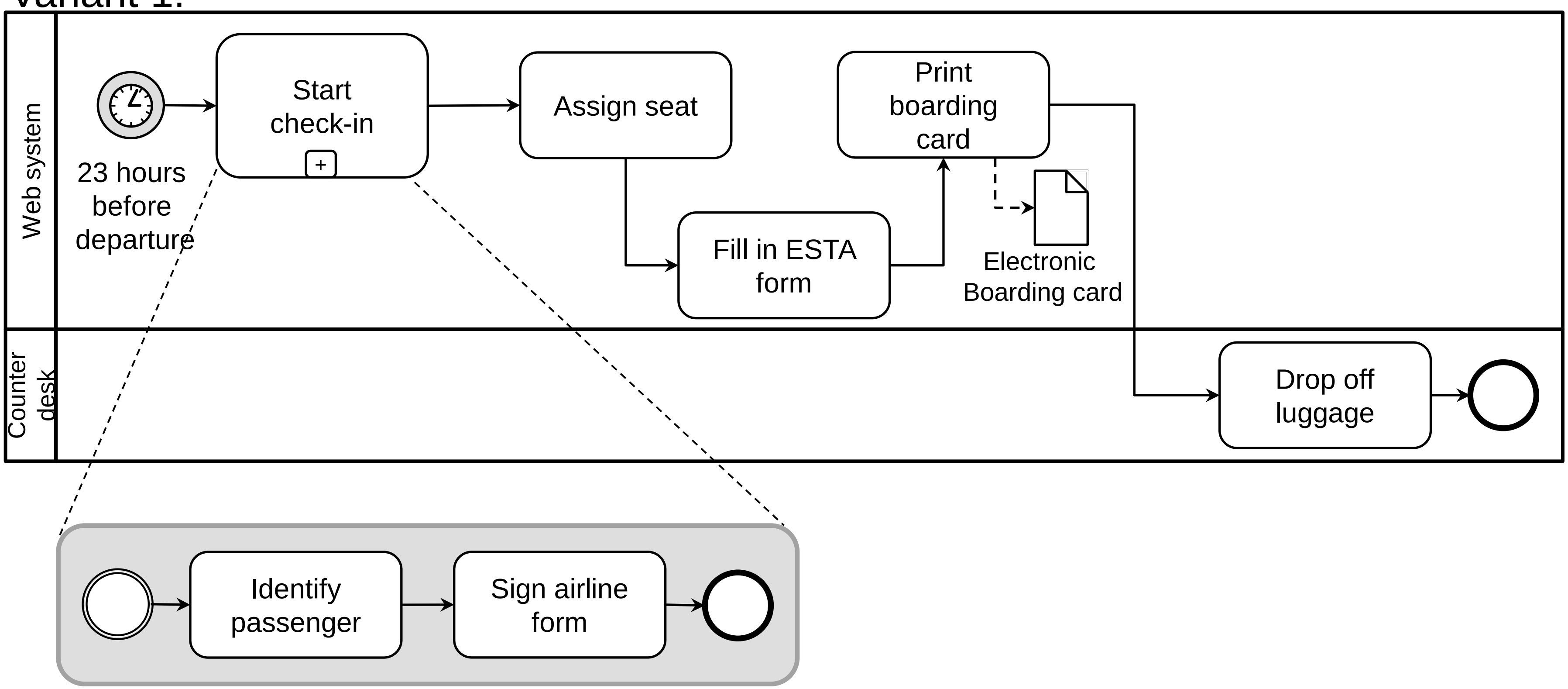
- Conditional branching,
- Separate process models, or
- Sub-processes

... for such purpose?

Limitations of Common BPMLs
Sub-processes

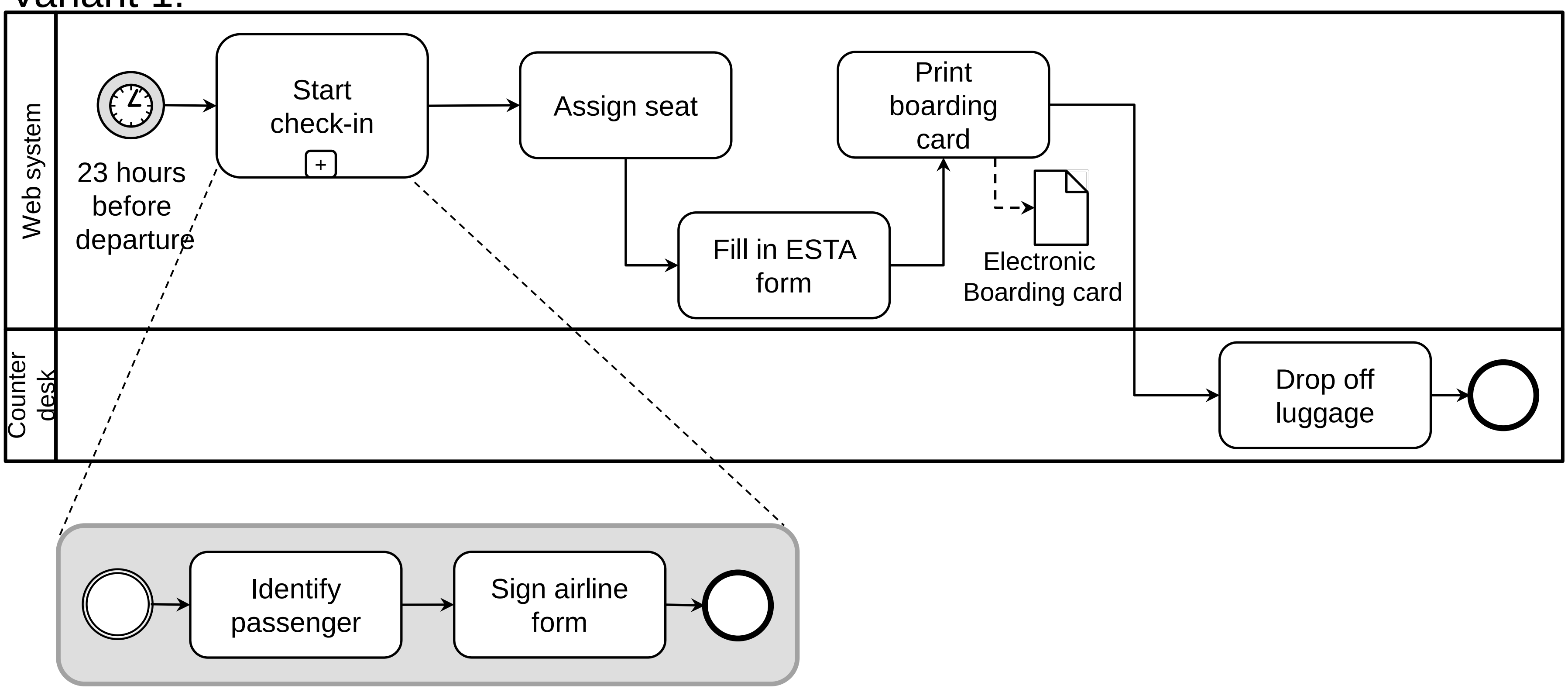
- (+) Promotes the reuse of common fragments
- (+) Allows reducing the size of the model

Variant 1:



- (-) Variability related issues cannot be explicitly represented

Variant 1:



Limitations of Common BPMLs

Even though these techniques are supported by commercial BPM tools, they do not enable transparent and explicit management of process variants

Variability in other Domains

- Is there something out in other domains that can be used?

Variability in the SPL domain

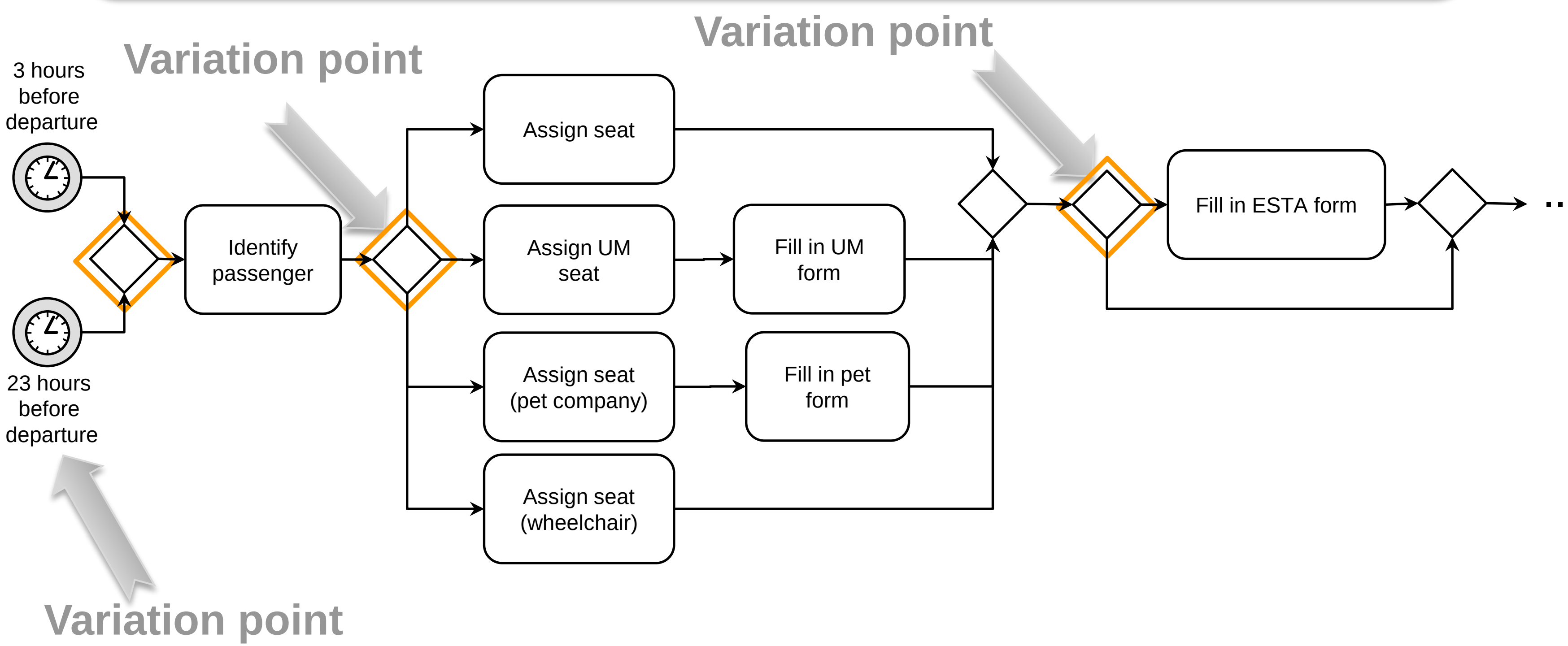
- Software Product Line (SPL) Engineering
  - Put emphasis on Reusability and Flexibility by
    - Consolidating and capitalizing on commonality through the product line
    - Focusing on product variations
  - Objectives:
    - Create a collection of similar software systems from a shared set of software assets using a common means of production

Language requirements

- Variation point
- Alternative process element
- Alternative process element context
- Alternative process element relationship
- Variation point resolution time

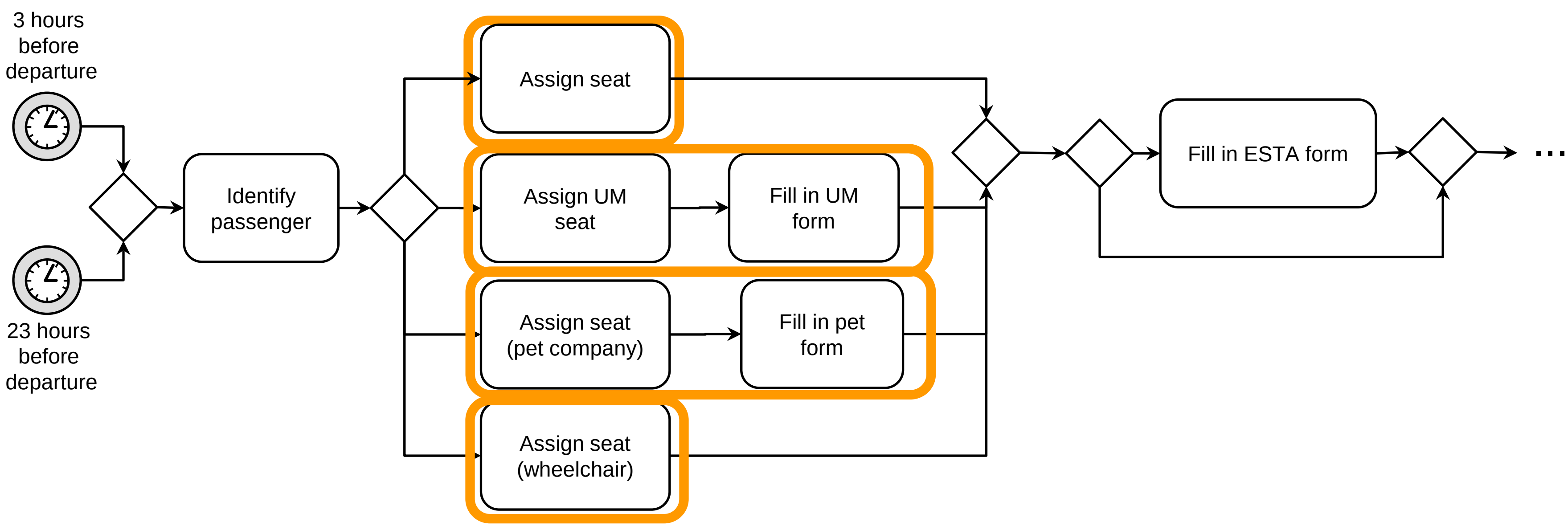
- Variation point

Precise position within a configurable process where different choices are possible depending on the current context or situation.



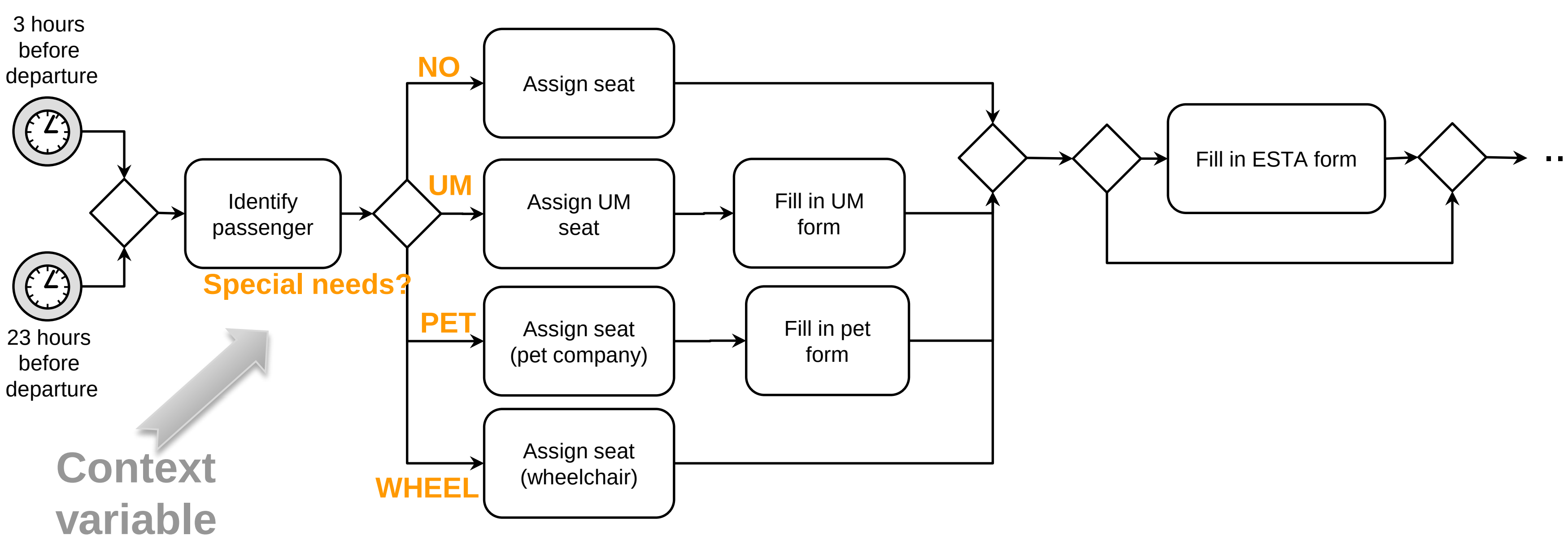
- Alternative process element

Particular option that may be instantiated at a specific *variation point* and may refer to any modelling element such as activities and their control flow, resources, data, events, or operations.



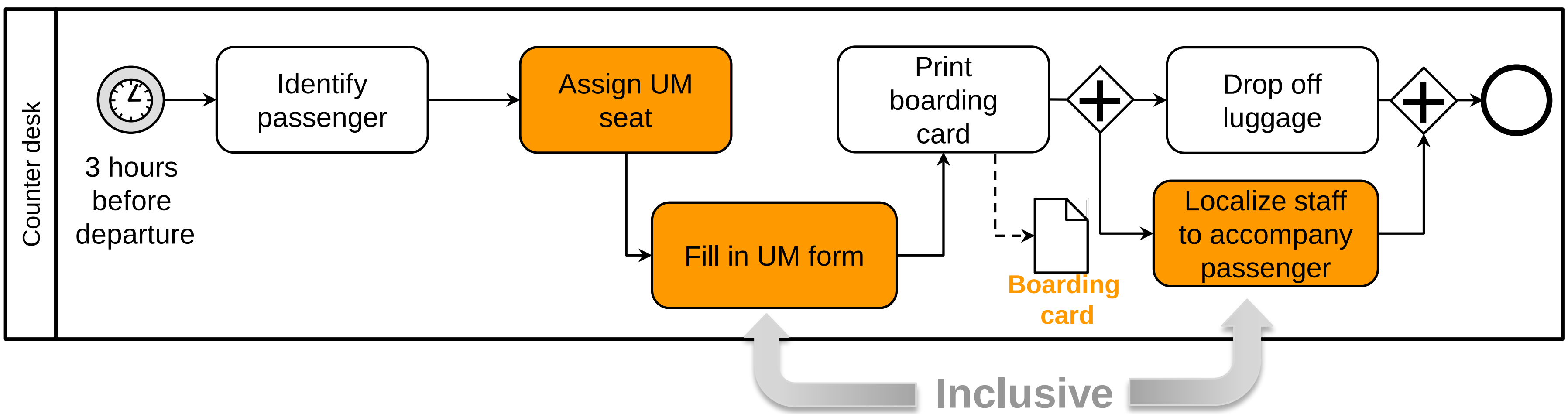
- Alternative process element context

Subset of process variables whose values make a particular *alternative process element* to become instantiated for a *variation point.*



- Alternative process element relationship

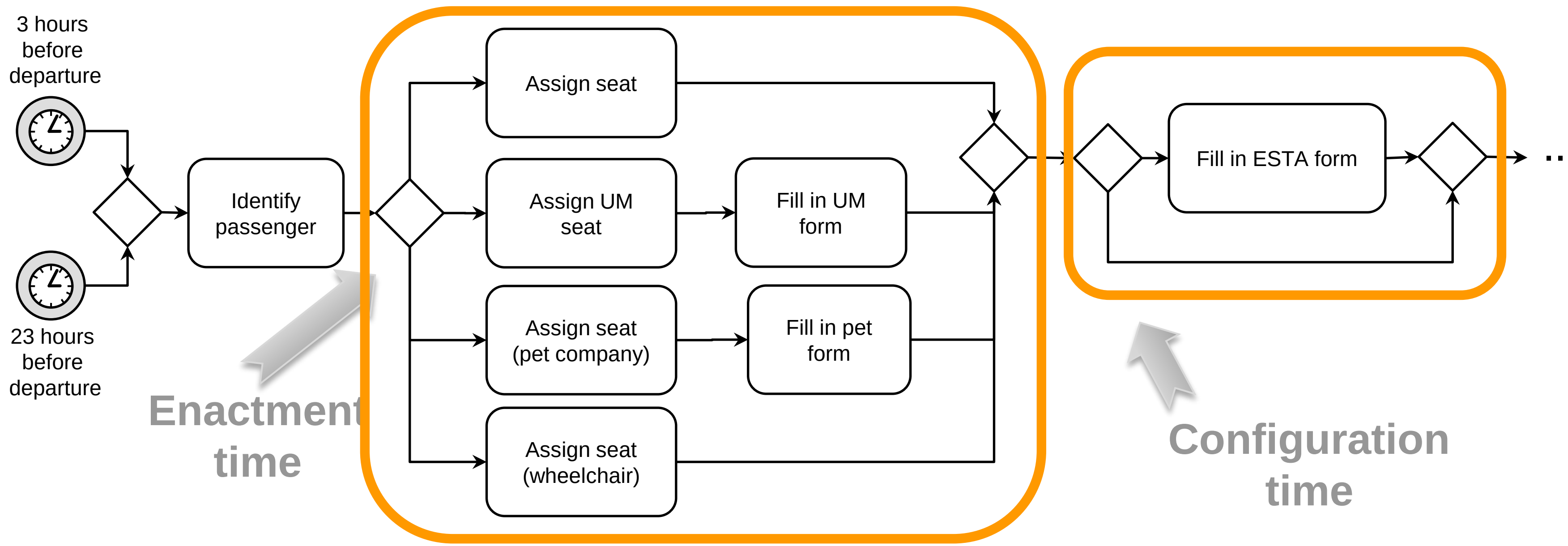
Constraint of use between two or more *alternative process elements*. These constraints are defined based on semantic relationships to ensure the proper use of the involved *alternative process elements* within a specific context.



Language requirements

- Variation point resolution time

This requirement should allow modellers to distinguish between variation points whose resolution depends on the initial context (configuration time) or on the current context of a process instance (enactment time).



Representing BP Variability

- Two main approaches:
  - Behavioural-based approaches
  - Structural-based approaches

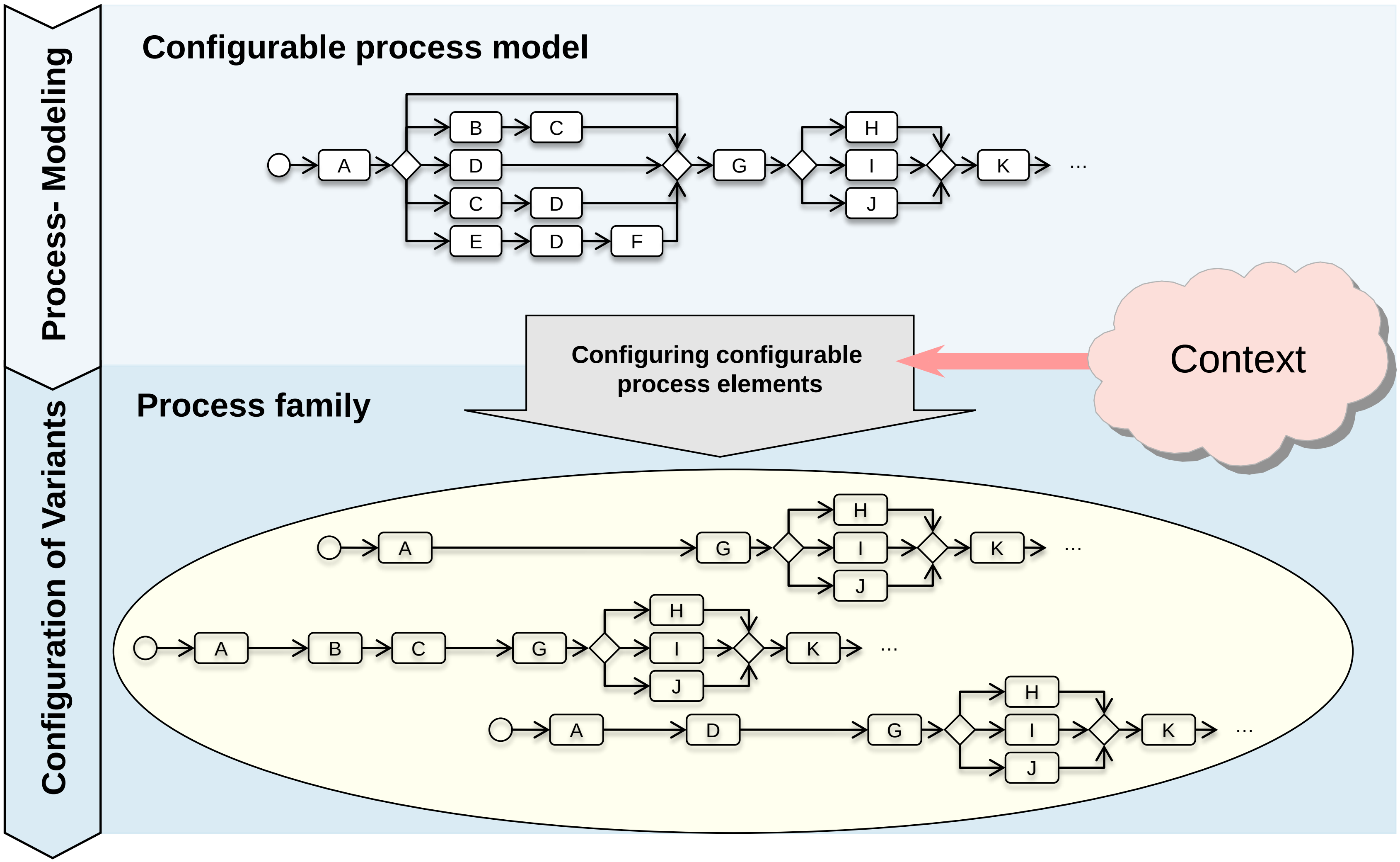
Representing BP Variability

- Two main approaches:
  - Behavioural-based approaches
  - Structural-based approaches

Representing BP Variability
Behavioural-based approaches

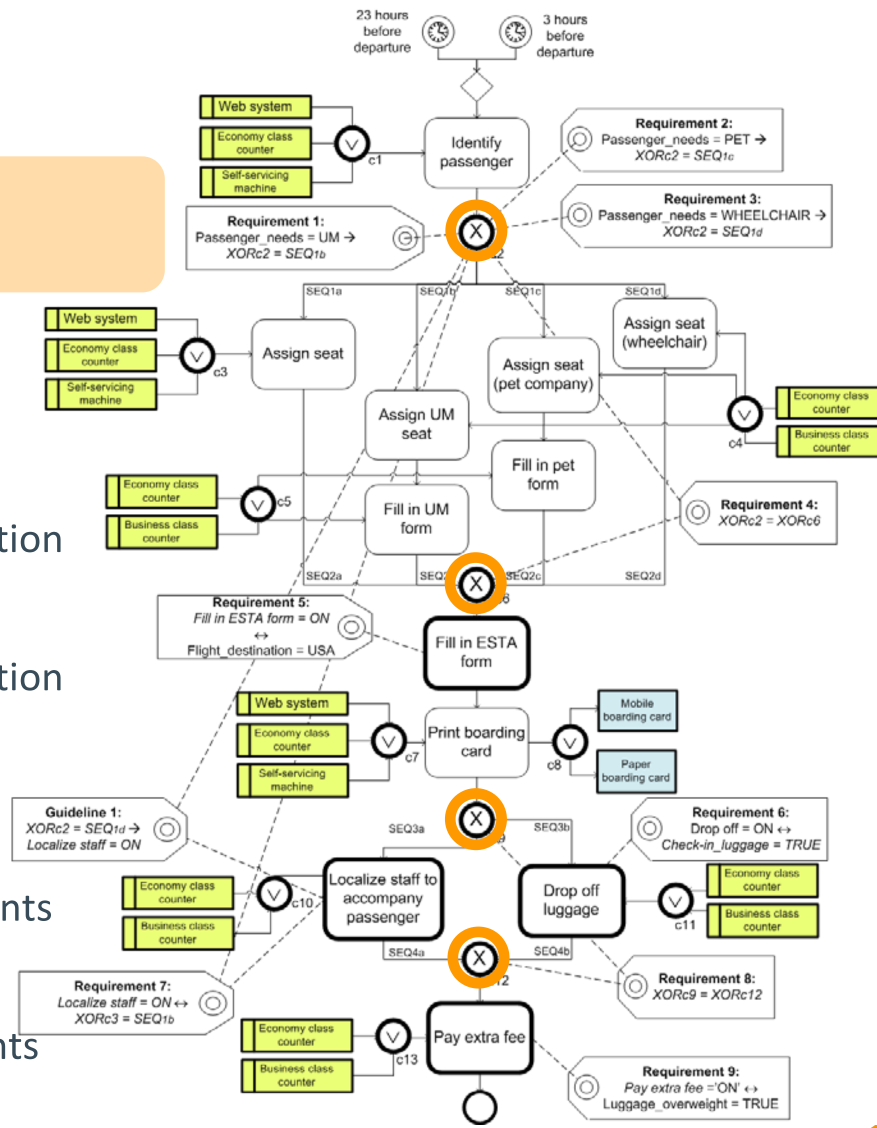
- Behavioural-based approaches
  - Single modelling artefact (Configurable process model)
  - Process variant derivation by removing parts from the configurable process model
    - Techniques for variant derivation:
      - Configurable nodes and configuration requirements and guidelines
      - Hiding and blocking
  - Proposals found in BPM literature
    - C-EPC/C-iEPC (Rosemann et al. 2007/La Rosa et al. 2008)
    - C-YAWL (Gottschalk et al. 2008)

Representing BP Variability
Behavioural-based approaches



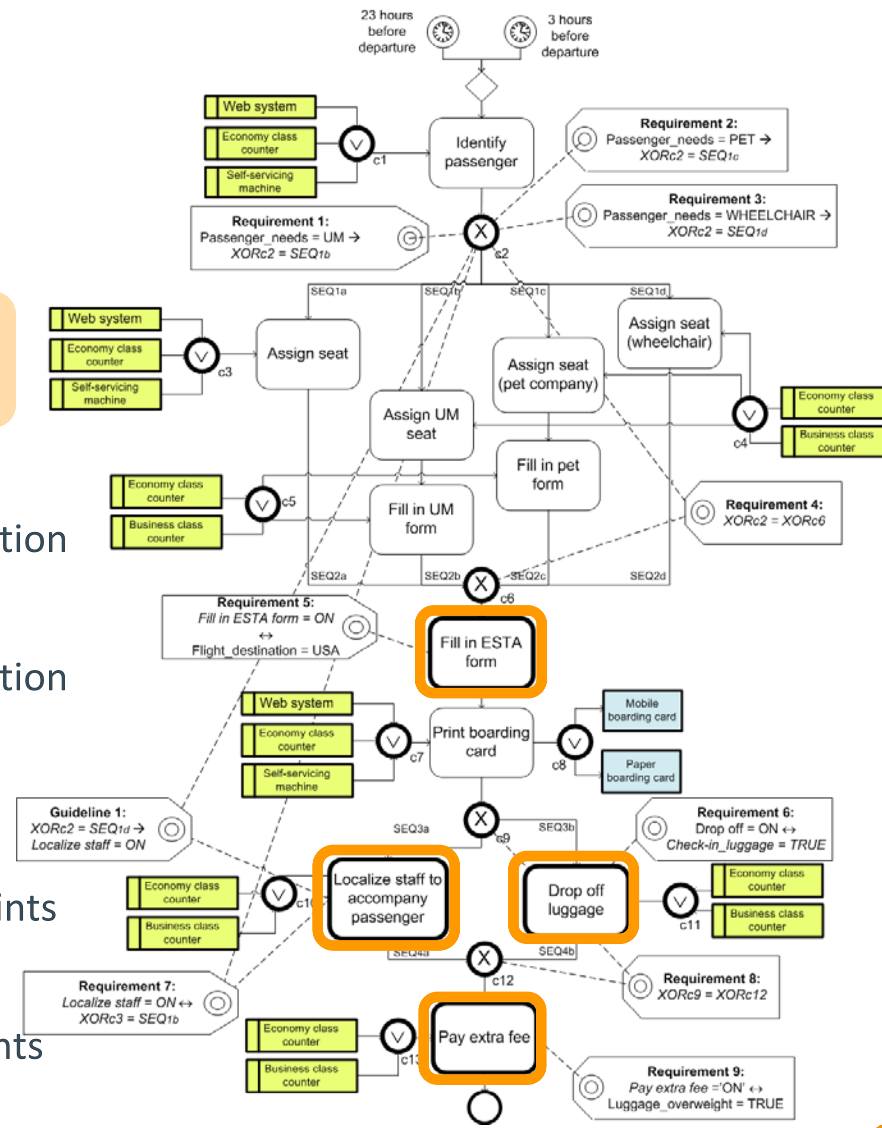
Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints



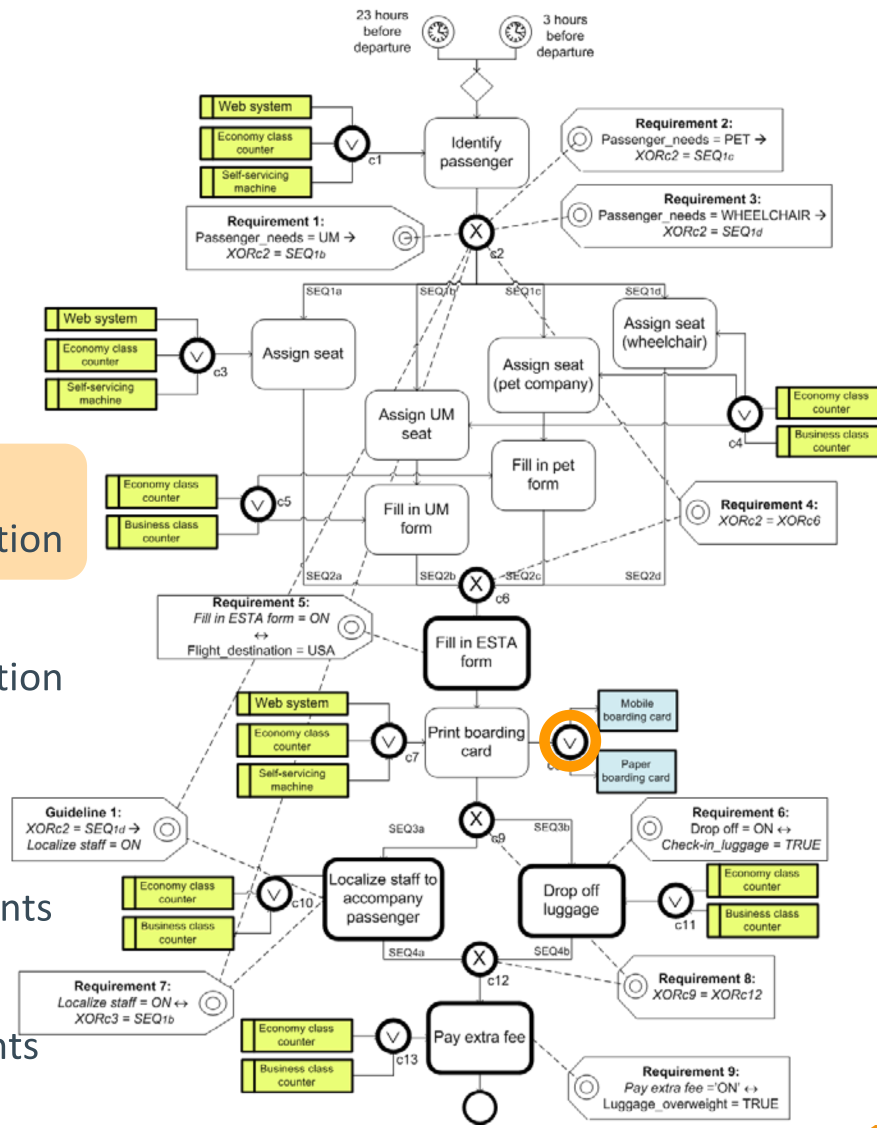
Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints



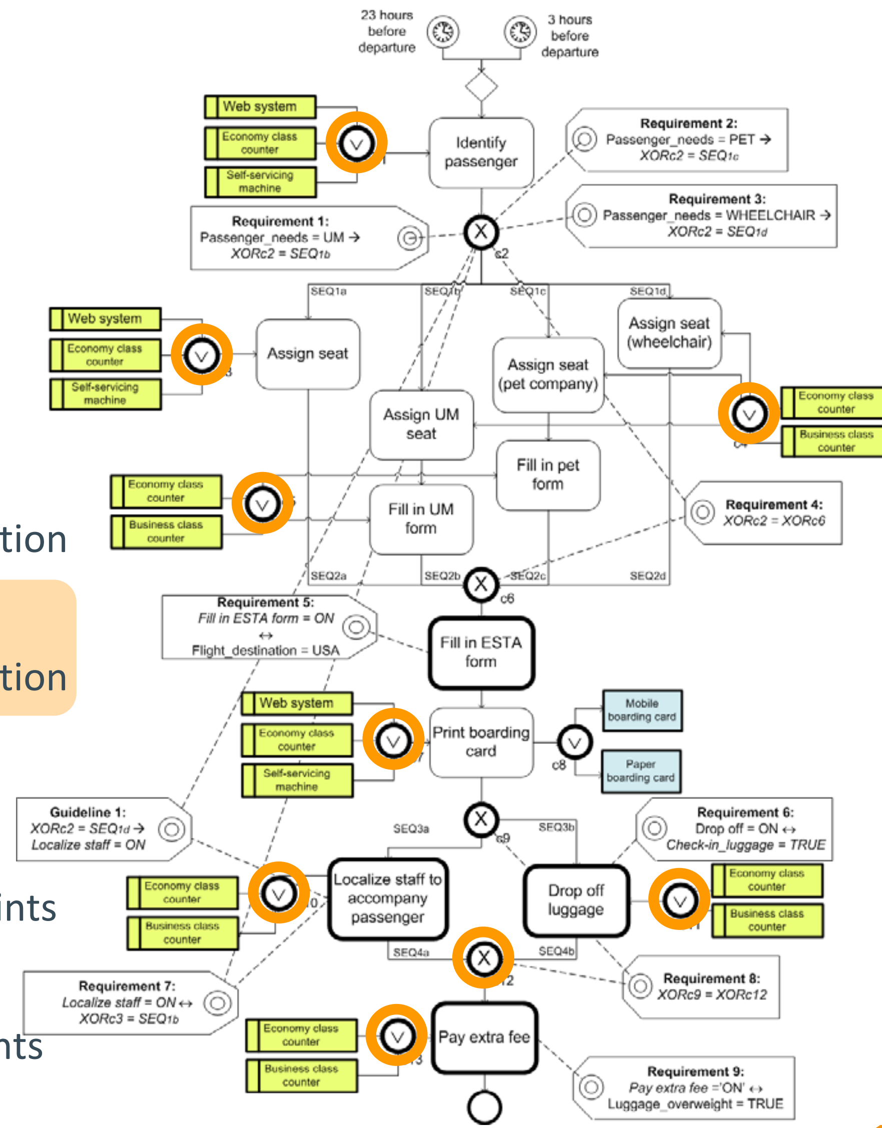
Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints



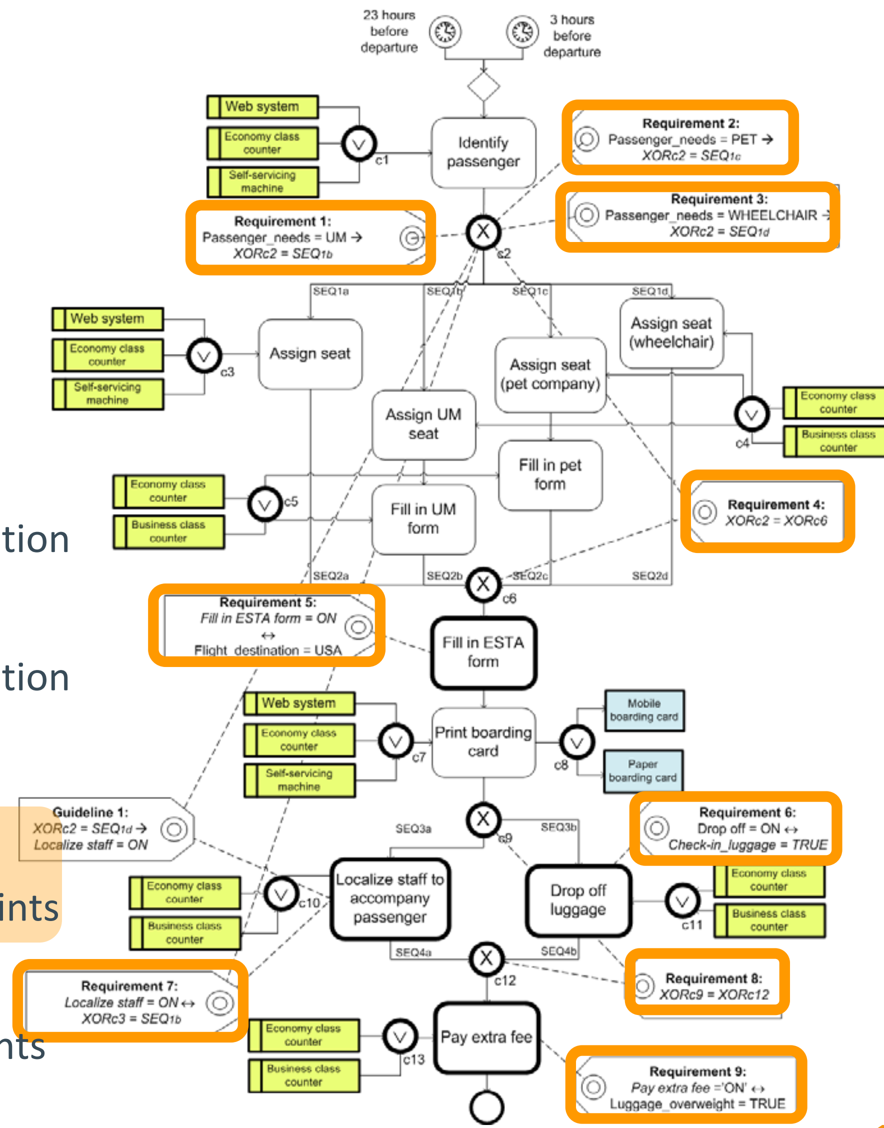
Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints



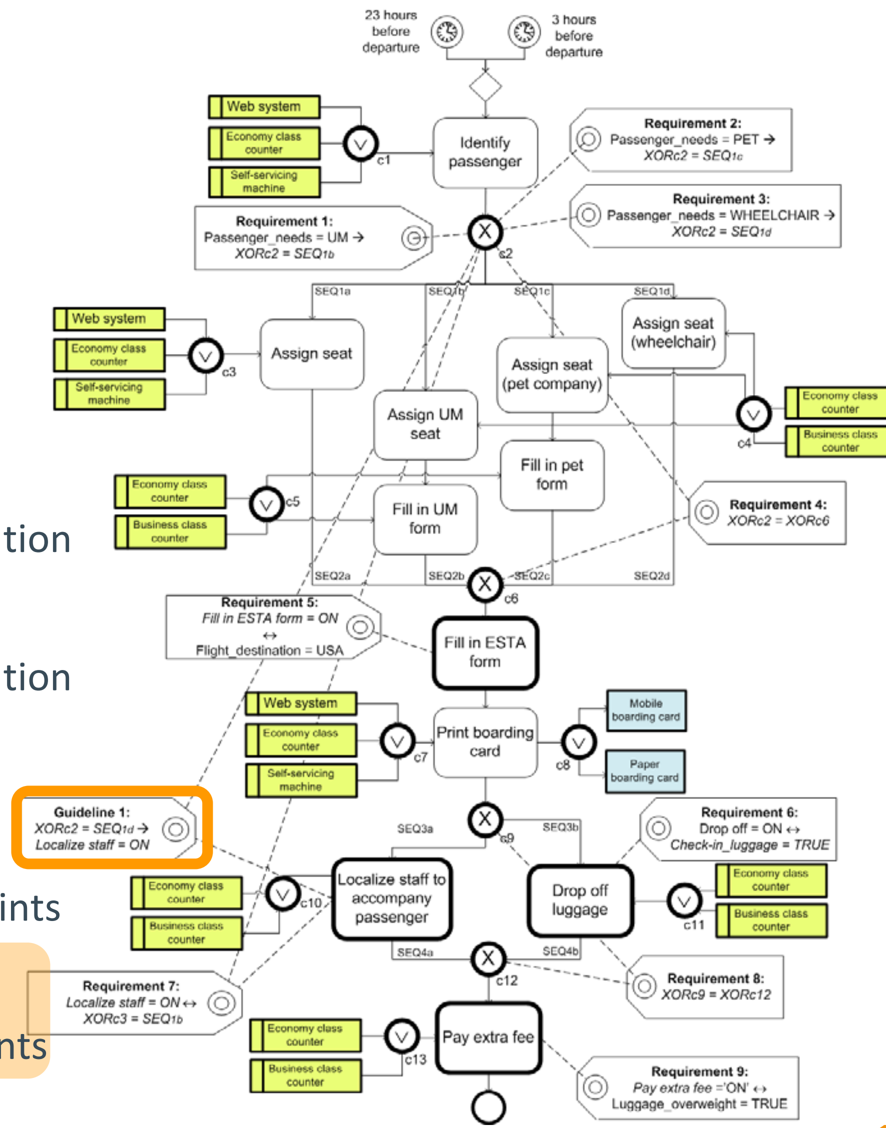
Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints



Representing BP Variability
Behavioural-based approaches

- C-iEPC
  - Configurable nodes
    - Connectors
      - Restrict their behaviour or SEQ
    - Functions
      - Configured as ON, OFF, or OPT
    - Objects
      - 2 Dim: Optionality and specialization
    - Roles
      - 2 Dim: Optionality and specialization
  - Configuration...
    - Requirements
      - State hard configuration constraints
    - Guidelines
      - State soft configuration constraints

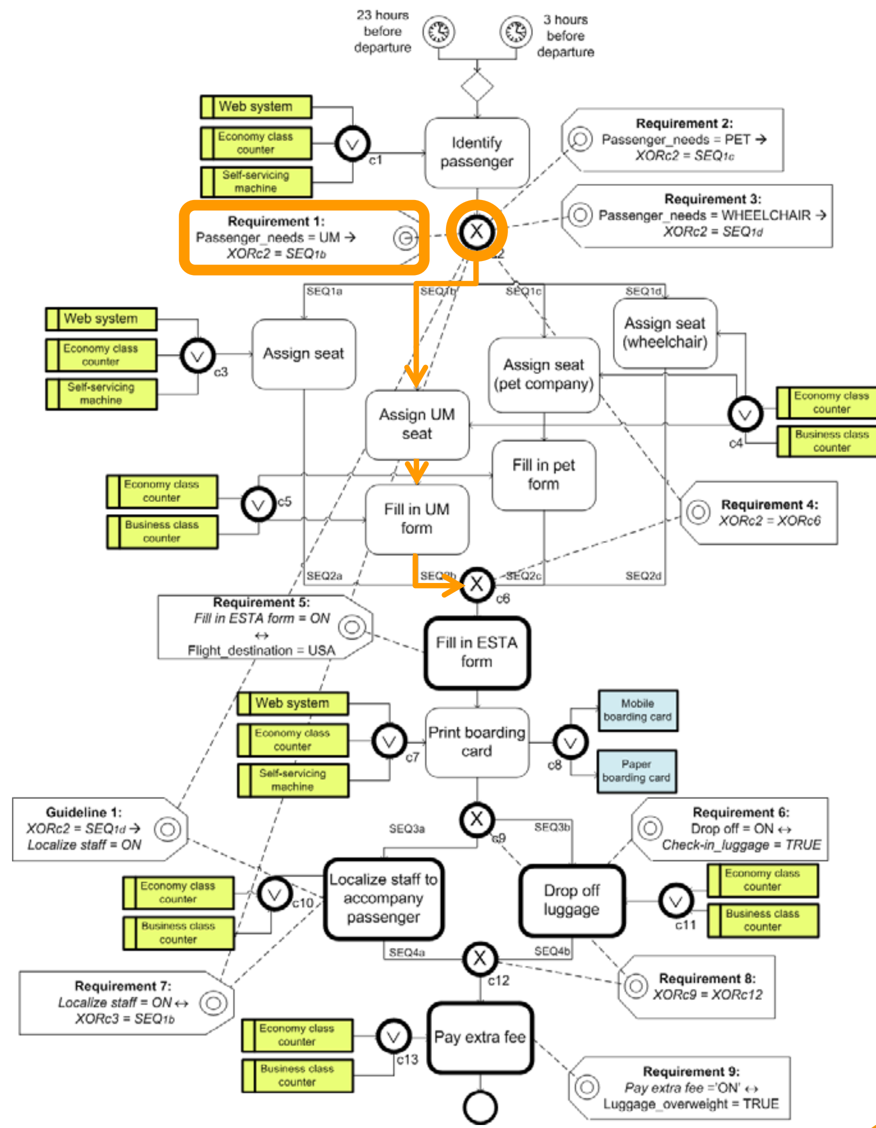


Understandability Task
Process Variant Extraction

- C-iEPC
  - Let’s derive the variant where:
    - Unaccompanied Minor (UM)
    - Check-in luggage
    - No luggage overweight

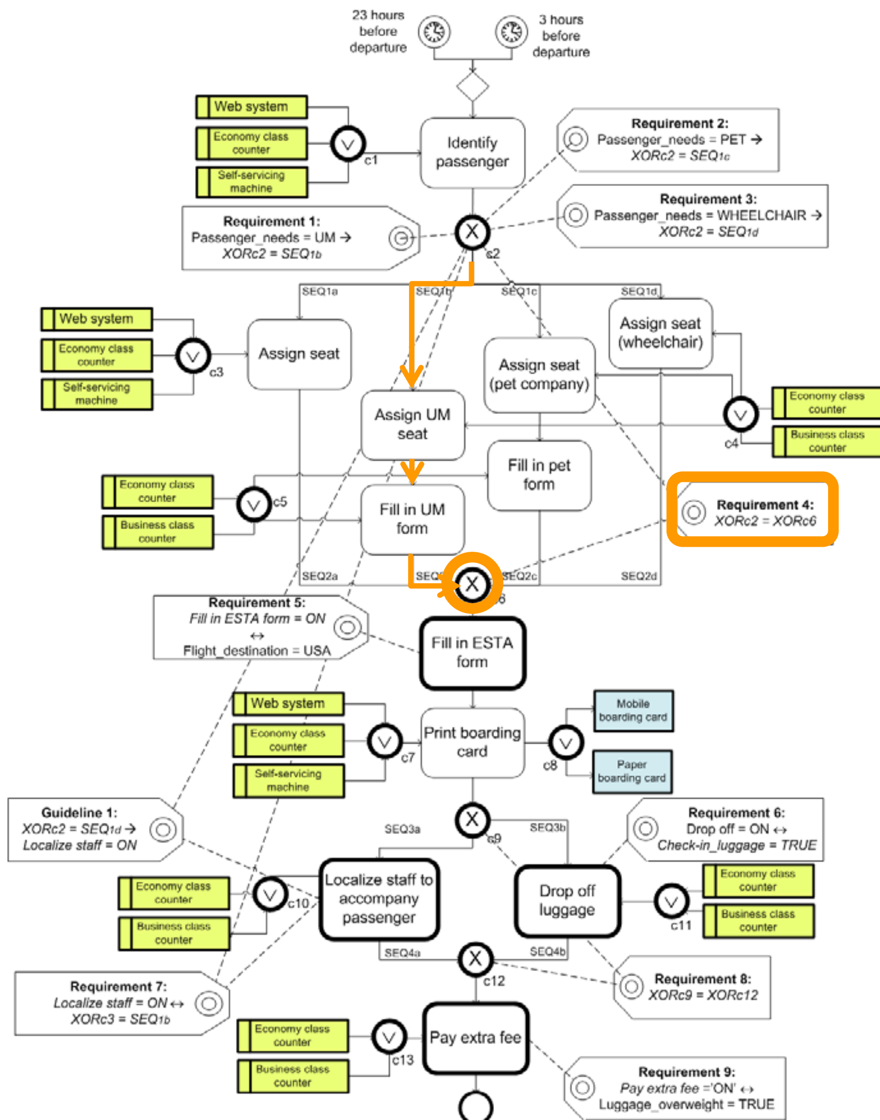
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b



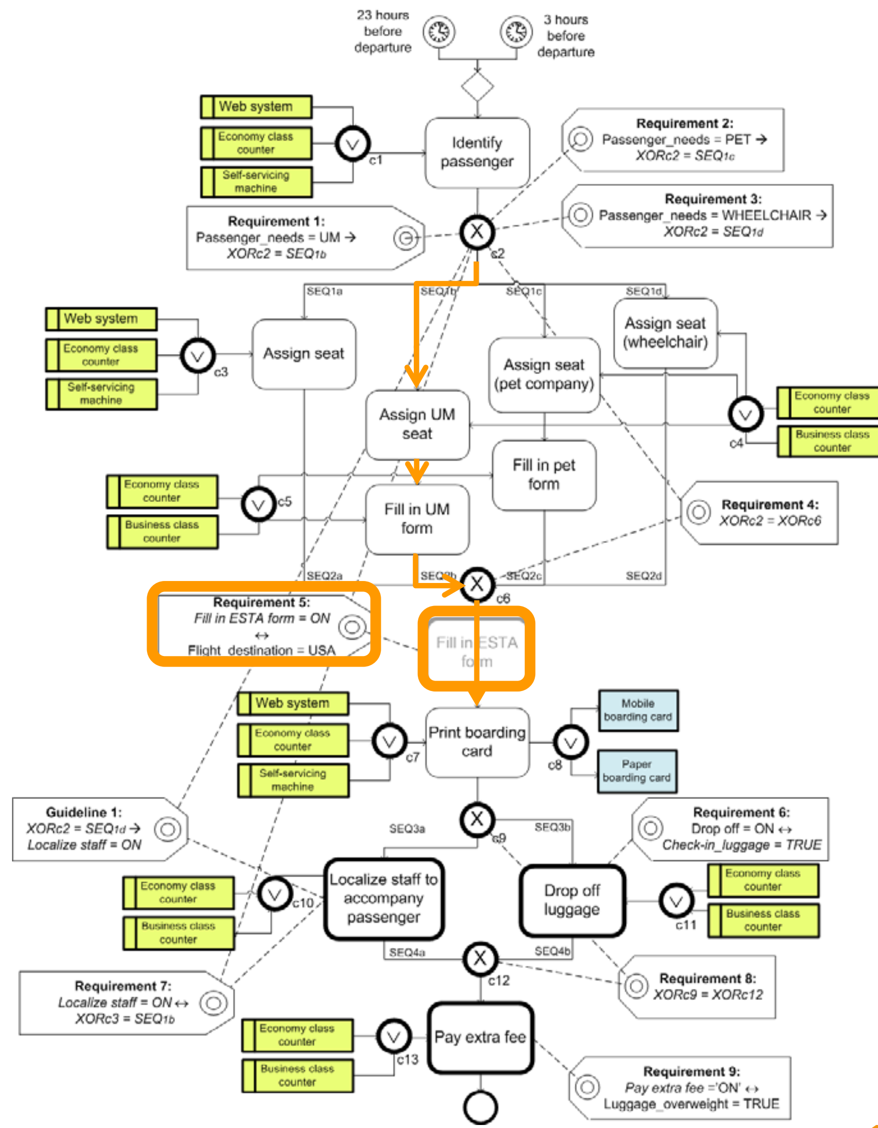
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b



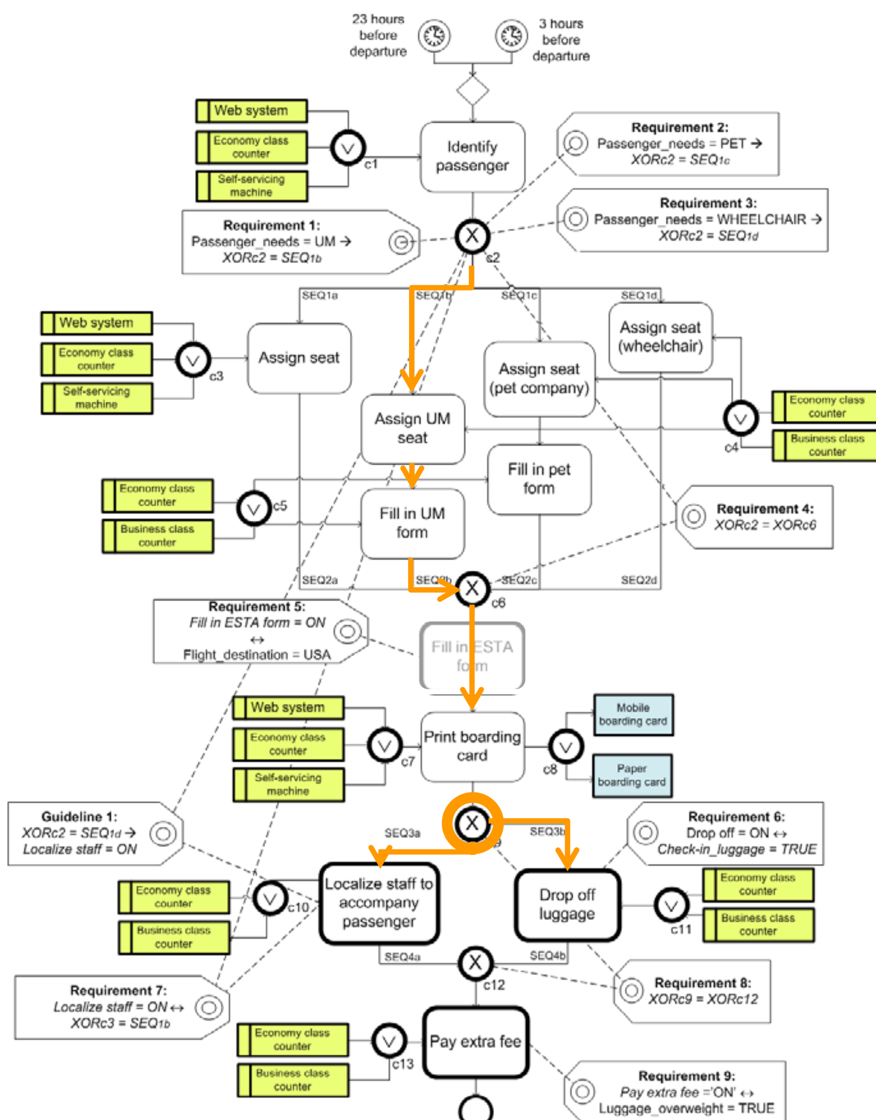
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
  - Functions
    - Fill in ESTA form = OFF



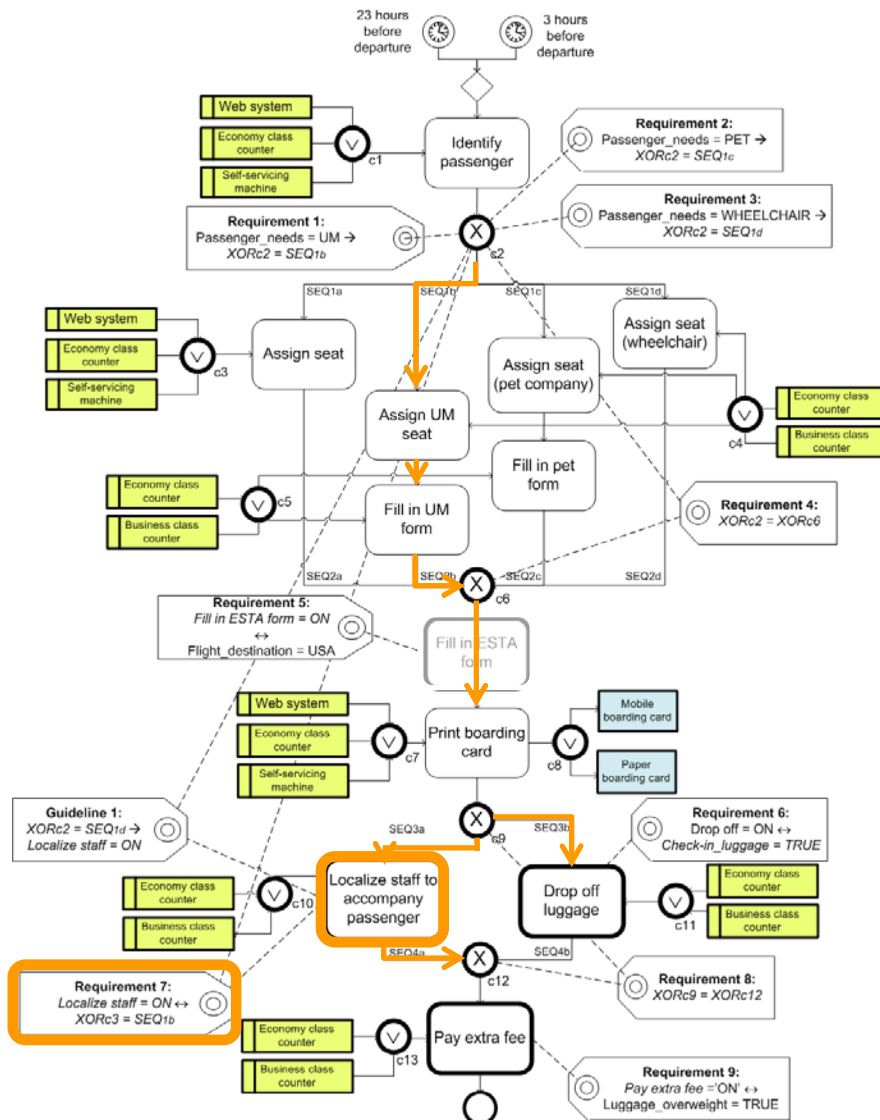
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
    - XORc9 = OR
  - Functions
    - Fill in ESTA form = OFF



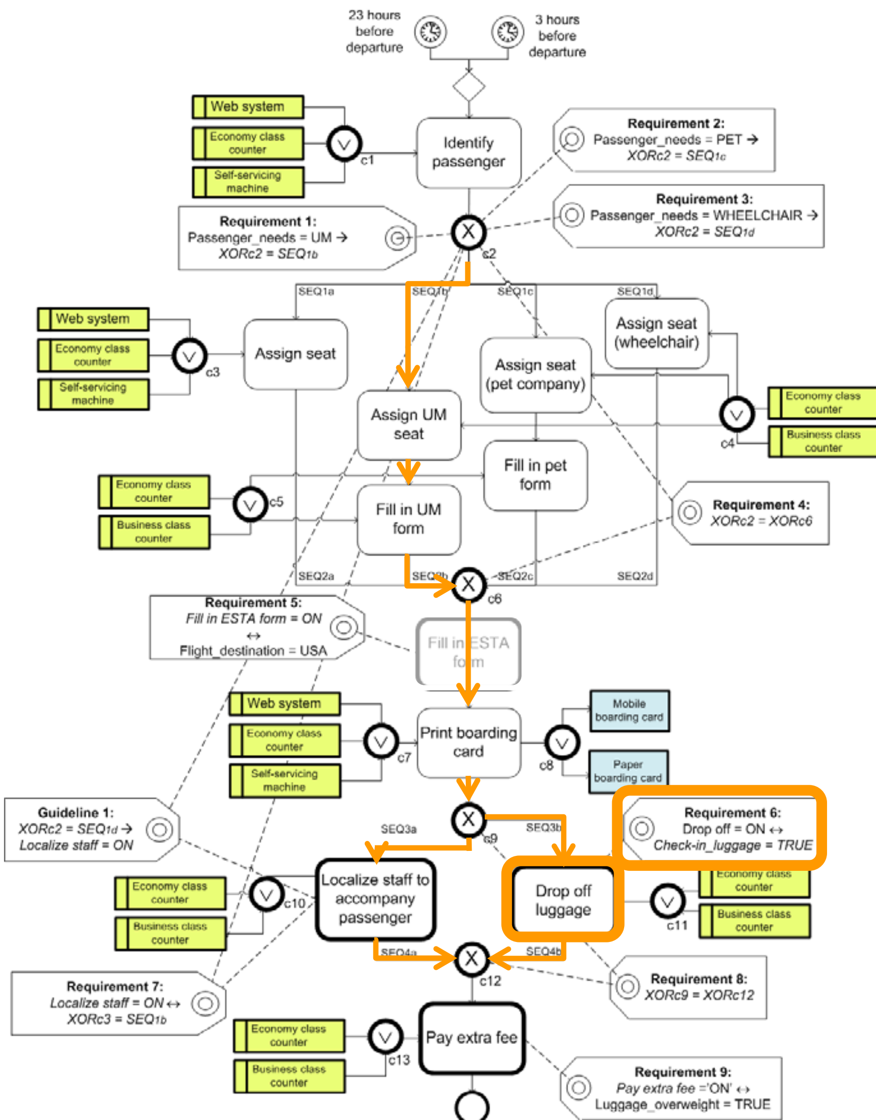
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
    - XORc9 = OR
  - Functions
    - Fill in ESTA form = OFF
    - Localize staff = ON



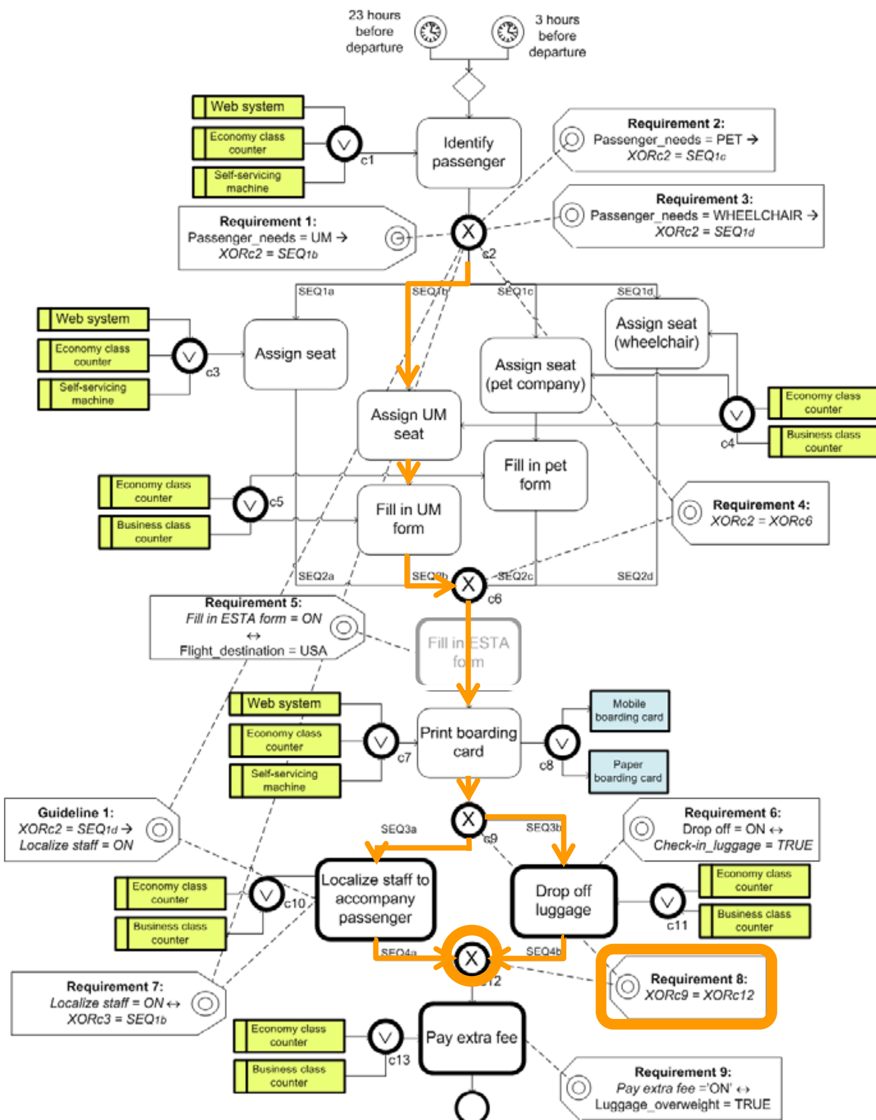
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
    - XORc9 = OR
  - Functions
    - Fill in ESTA form = OFF
    - Localize staff = ON
    - Drop off lugg. = ON



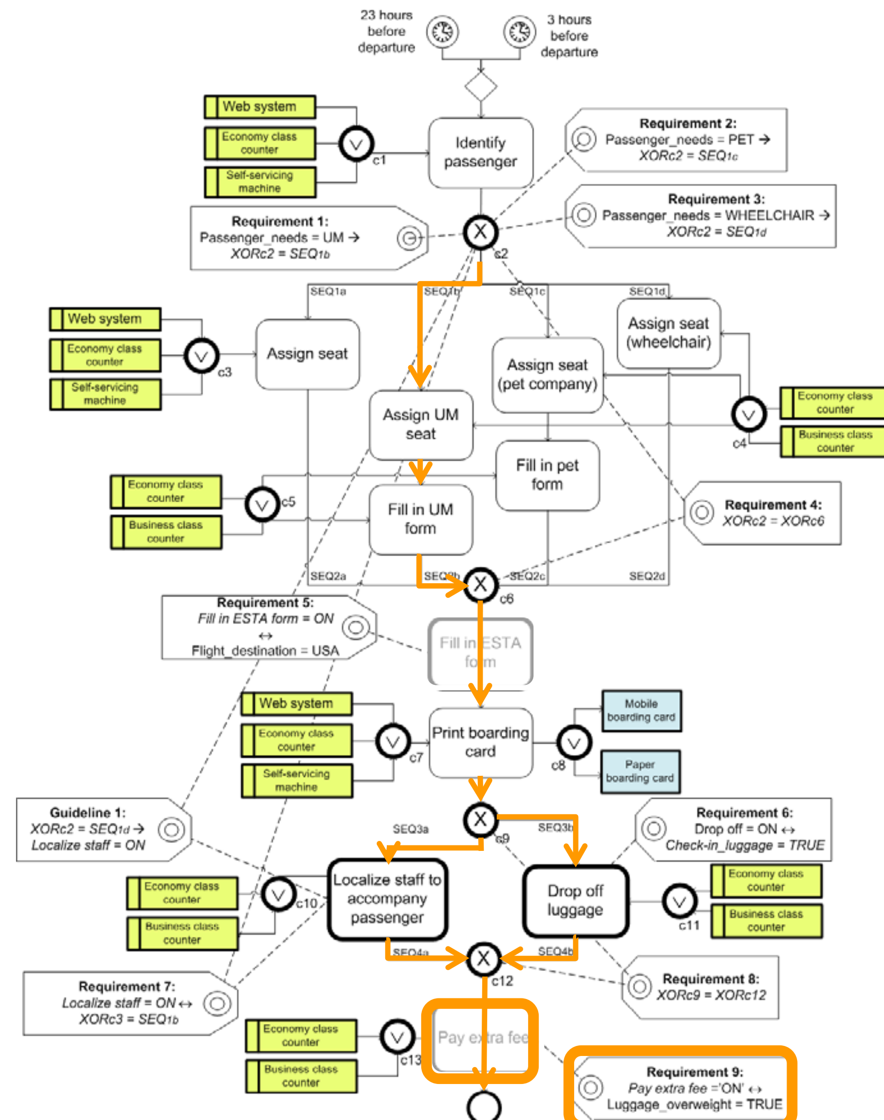
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
    - XORc9 = OR
    - XORc12 = OR
  - Functions
    - Fill in ESTA form = OFF
    - Localize staff = ON
    - Drop off lugg. = ON



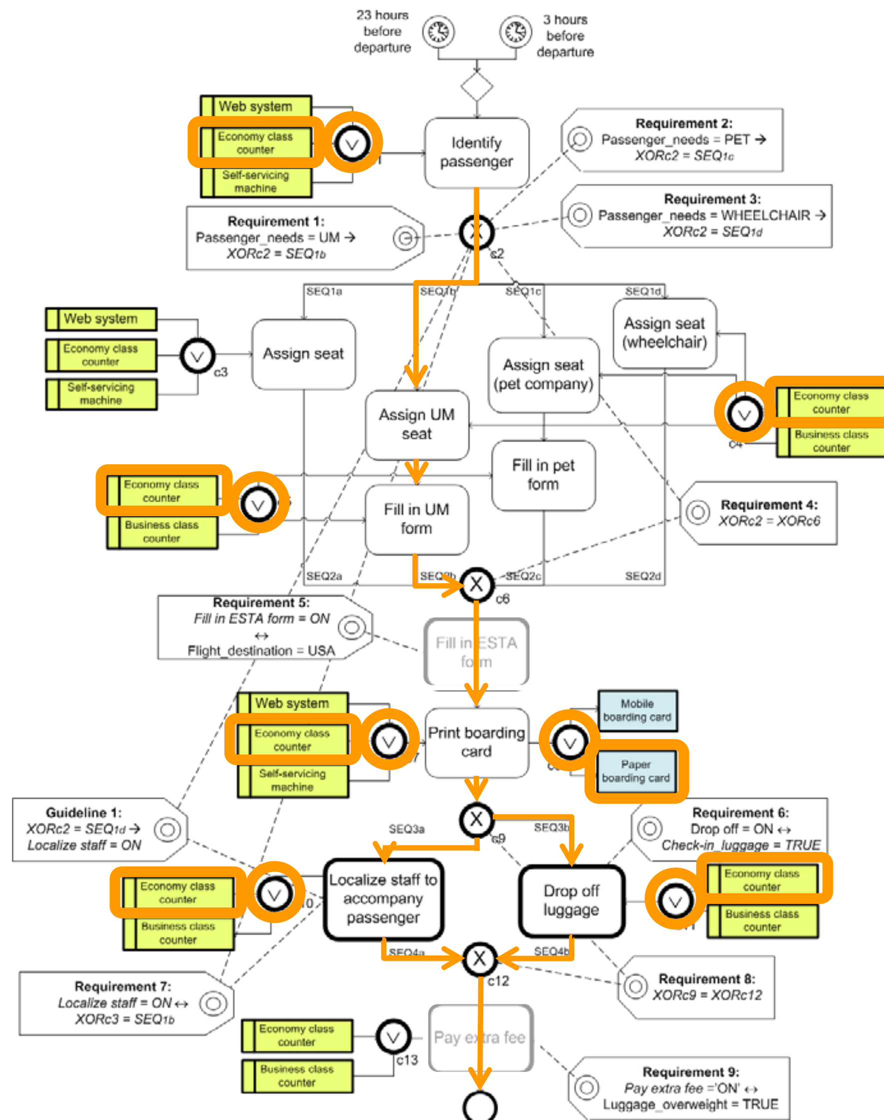
Understandability Task
Process Variant Extraction

- C-iEPC
  - Connectors
    - XORc2 = SEQ1b
    - XORc6 = SEQ2b
    - XORc9 = OR
    - XORc12 = OR
  - Functions
    - Fill in ESTA form = OFF
    - Localize staff = ON
    - Drop off lugg. = ON
    - Pay extra fee = OFF



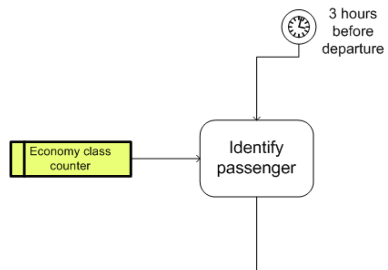
Understandability Task
Process Variant Extraction

- C-iEPC
  - Roles
    - c1 = Economy class counter
    - c4 = Economy class counter
    - c5 = Economy class counter
    - c7 = Economy class counter
    - c10 = Economy class counter
    - c11 = Economy class counter
  - Objects
    - C8 = Paper boarding card



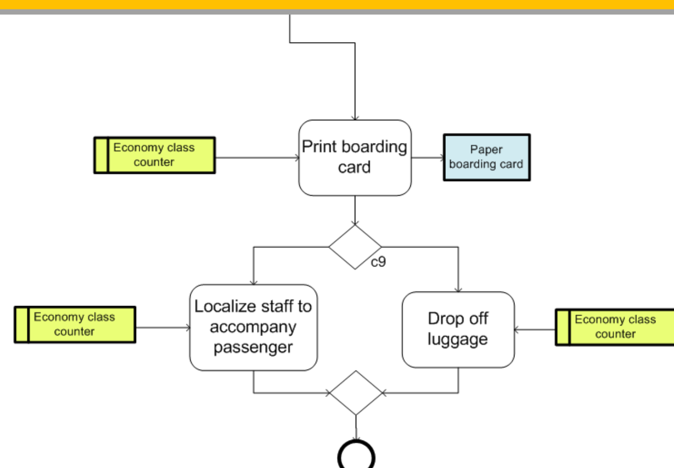
Understandability Task
Process Variant Extraction

- C-iEPC



Mental process followed in C-EPC:

1. Inspect all configurable nodes
2. Evaluate the associated requirements
3. Configure configurable nodes accordingly



- C-iEPC
  - Cognitive discussion
    - 3 basic operations:
      - Locating elements
        - » Easy to perform thanks to their visual differences
      - Evaluating Boolean expressions
        - » Can be pretty challenging depending on the complexity of the expression
        - » Some evaluations depend on previous decisions
      - Adapting the model accordingly
        - » Elements have to be mentally removed

⇒Complexity depends on how many requirements have to be analyzed

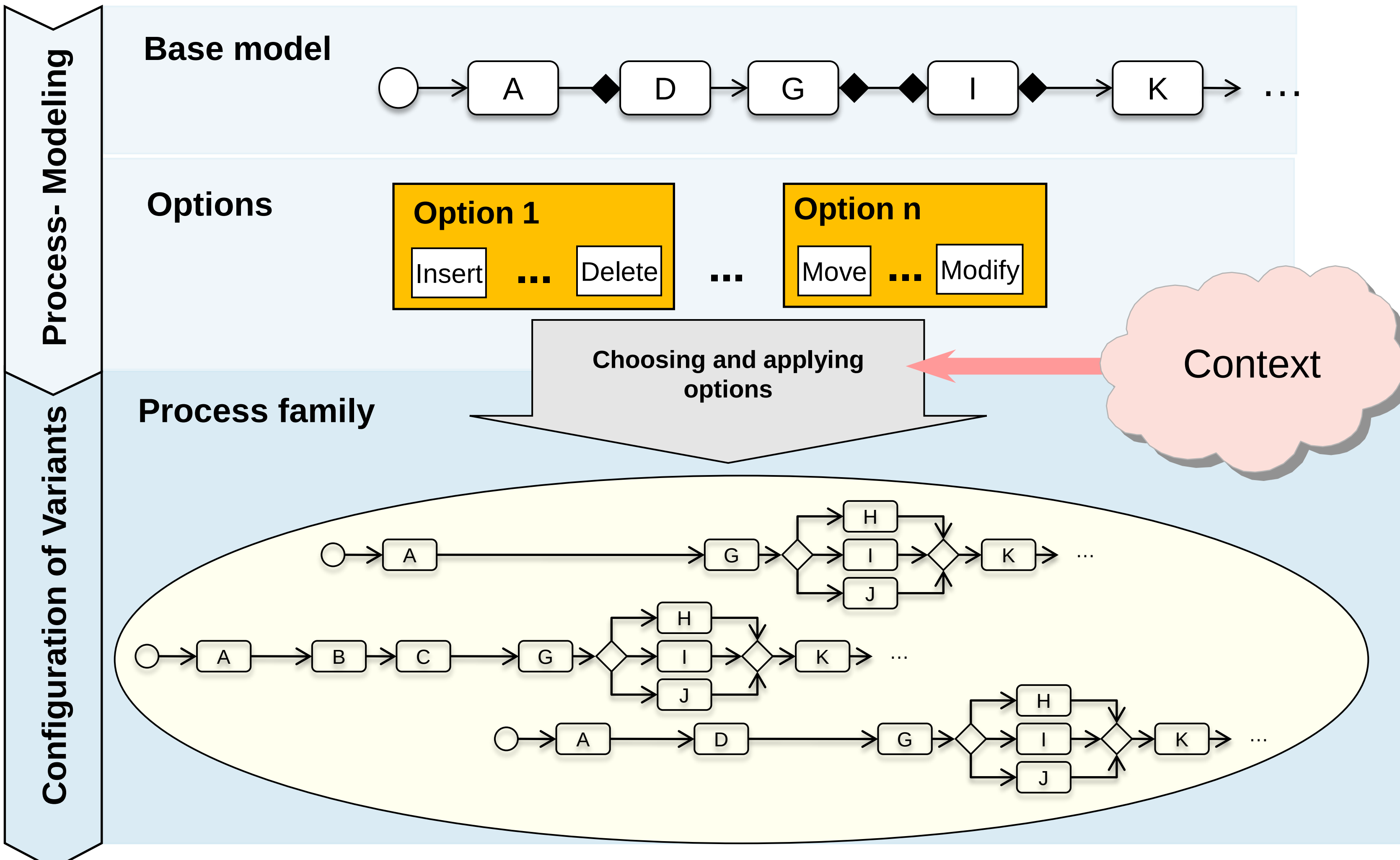
Representing BP Variability

- Two main approaches:
  - Behavioural-based approaches
  - Structural-based approaches

Representing BP Variability
Structural-based approaches

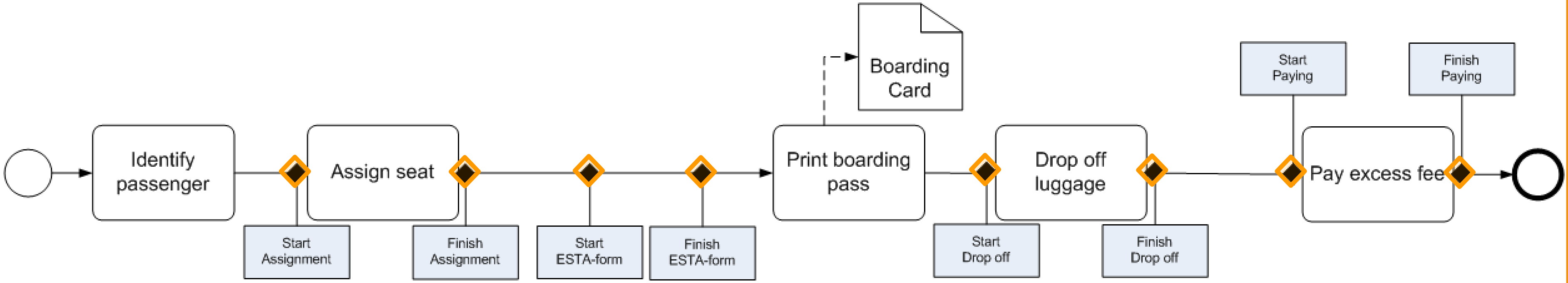
- Structural-based approaches
  - Several modelling artefacts
    - Base model
    - Change operations
  - Process variant derivation through the application of change operation to the base model
  - Proposals found in BPM literature
    - Provop (Hallerbach et al. 2010)
    - Rule representation and processing (Kumar and Wen 2012)

- Provop


Process- Modeling
Configuration of Variants
Base model
A
D
G
I
K
...
Options
Option 1
Insert
...
Delete
...
Option n
Move
...
Modify
Choosing and applying options
Context
Process family
A
G
H
I
J
K
...
A
B
C
G
H
I
J
K
...
A
D
G
H
I
J
K
...

- Provop
  - Base model
    - Policies:
      - Standard process
      - Most frequently used process
      - Minimal average distance
      - Superset of all process variants
      - Intersection of all process variants

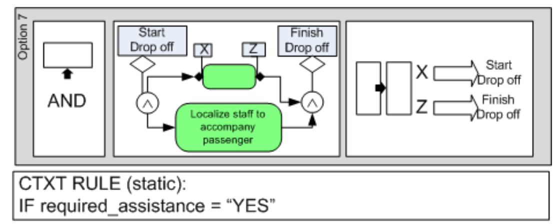
Variant: Passenger with no special needs, dropping off luggage, carrying luggage excess

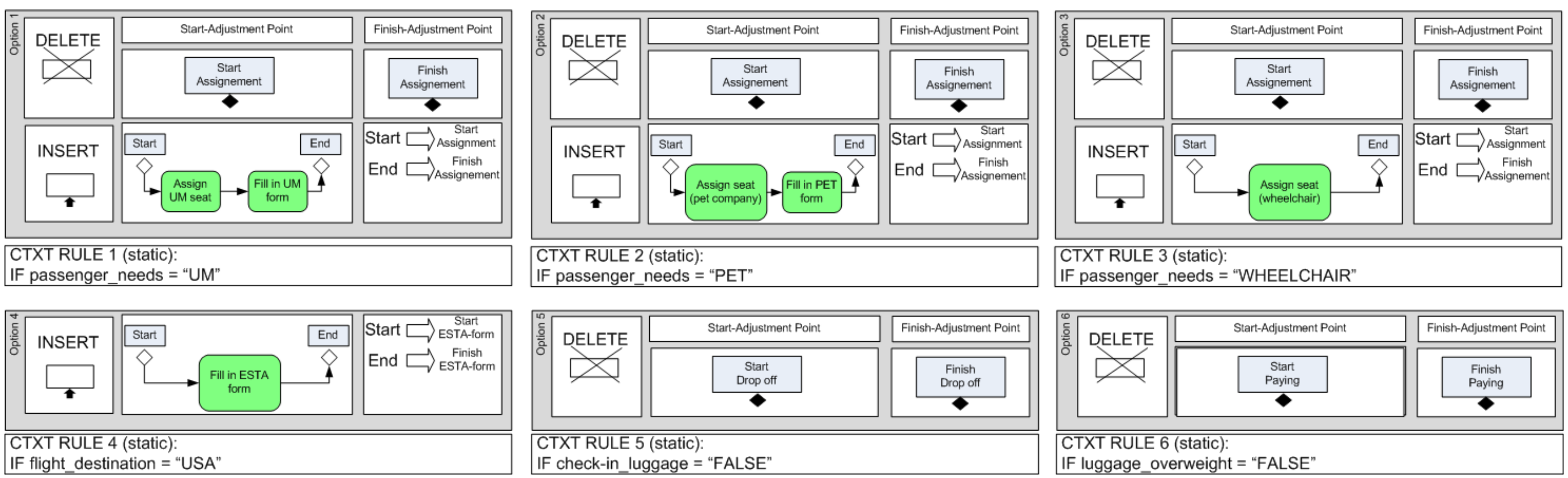


Adjustment Points

Representing BP Variability
Structural-based approaches

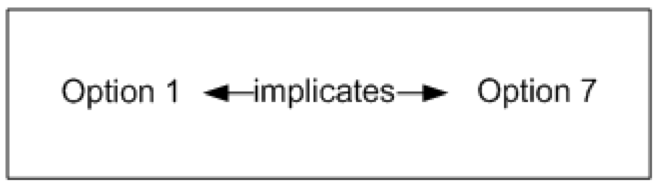
- Provop
  - Change Options
    - Allow deriving new variants
    - Are applied to the base model

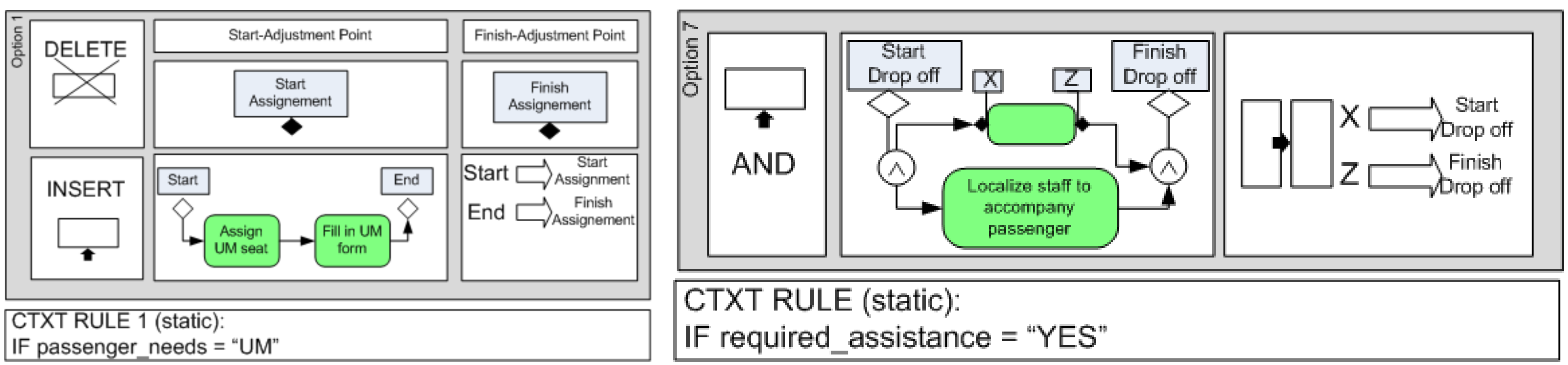




Representing BP Variability
Structural-based approaches

- Provop
  - Options Constraints
    - Relationships : Implication, Exclusion, Order, etc.





- Provop
  - Context Model

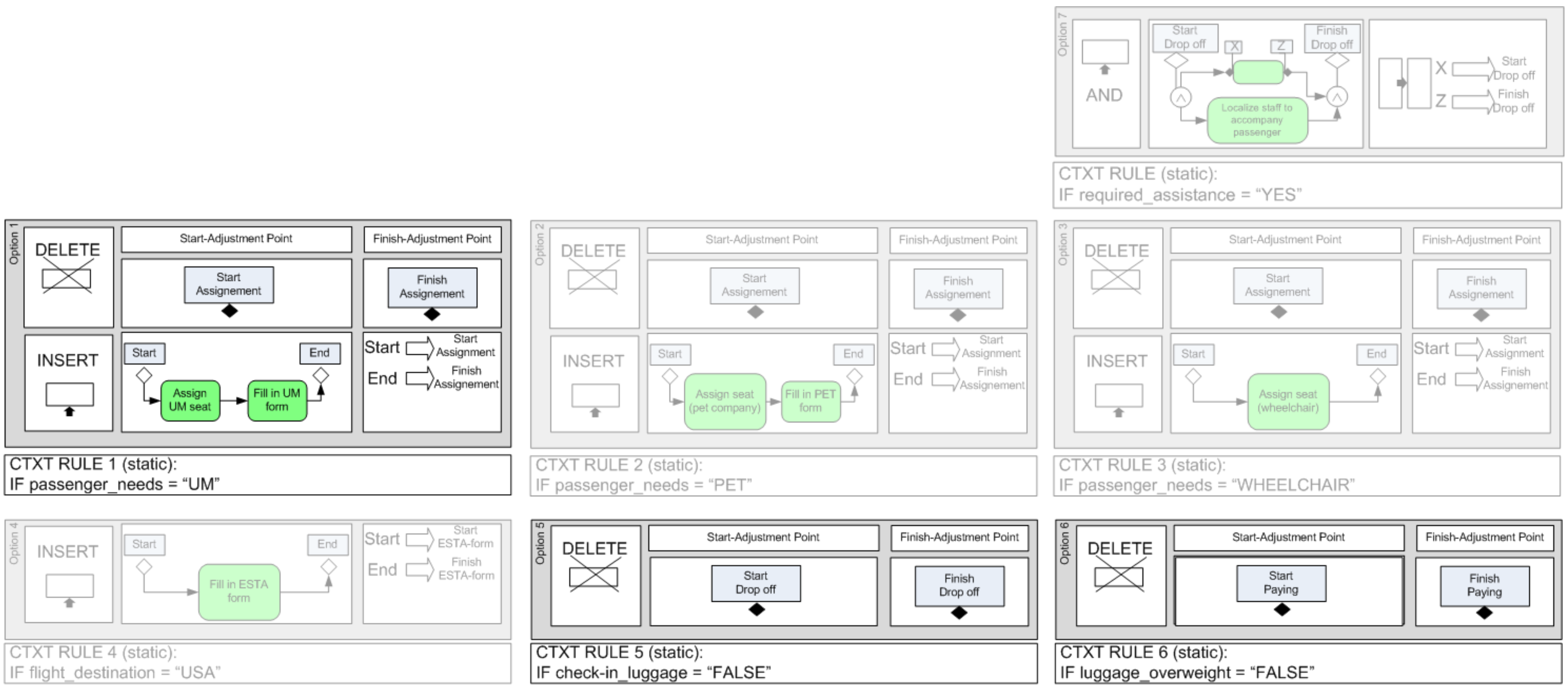
Context variable	Range of Values	Behaviour
Passenger_needs	NO, PET, UM, WHEELCHAIR	Static
Flight_destination	USA, EU	Static
Luggage_overweight	TRUE, FALSE	Static
Check-in_luggage	TRUE, FALSE	Static
Required_assistance	TRUE, FALSE	Static

Understandability Task
Process Variant Extraction

- Provop
  - Let’s derive the variant where:
    - Unaccompanied Minor (UM)
    - Check-in luggage
    - No luggage overweight

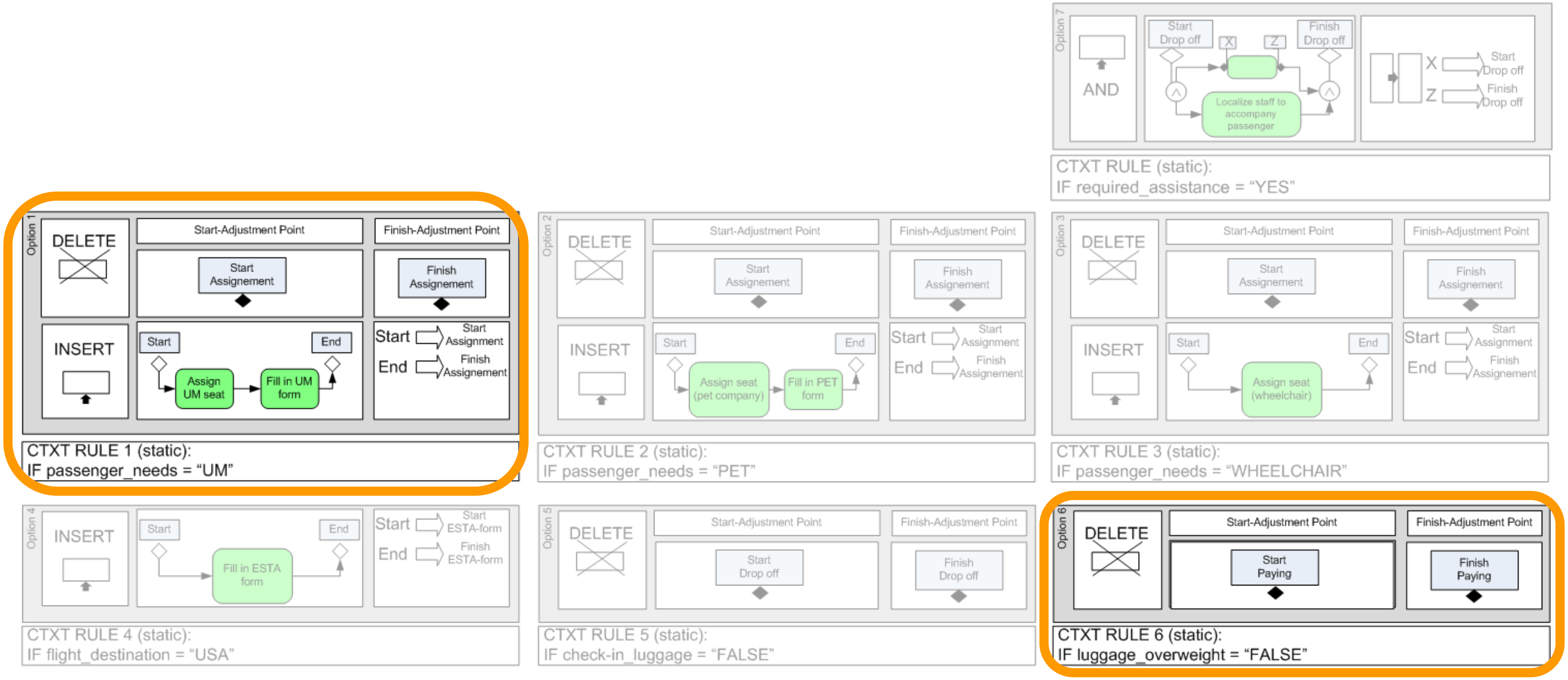
Understandability Task
Process Variant Extraction

- Provop
  1. Examine all change options and their Boolean expressions



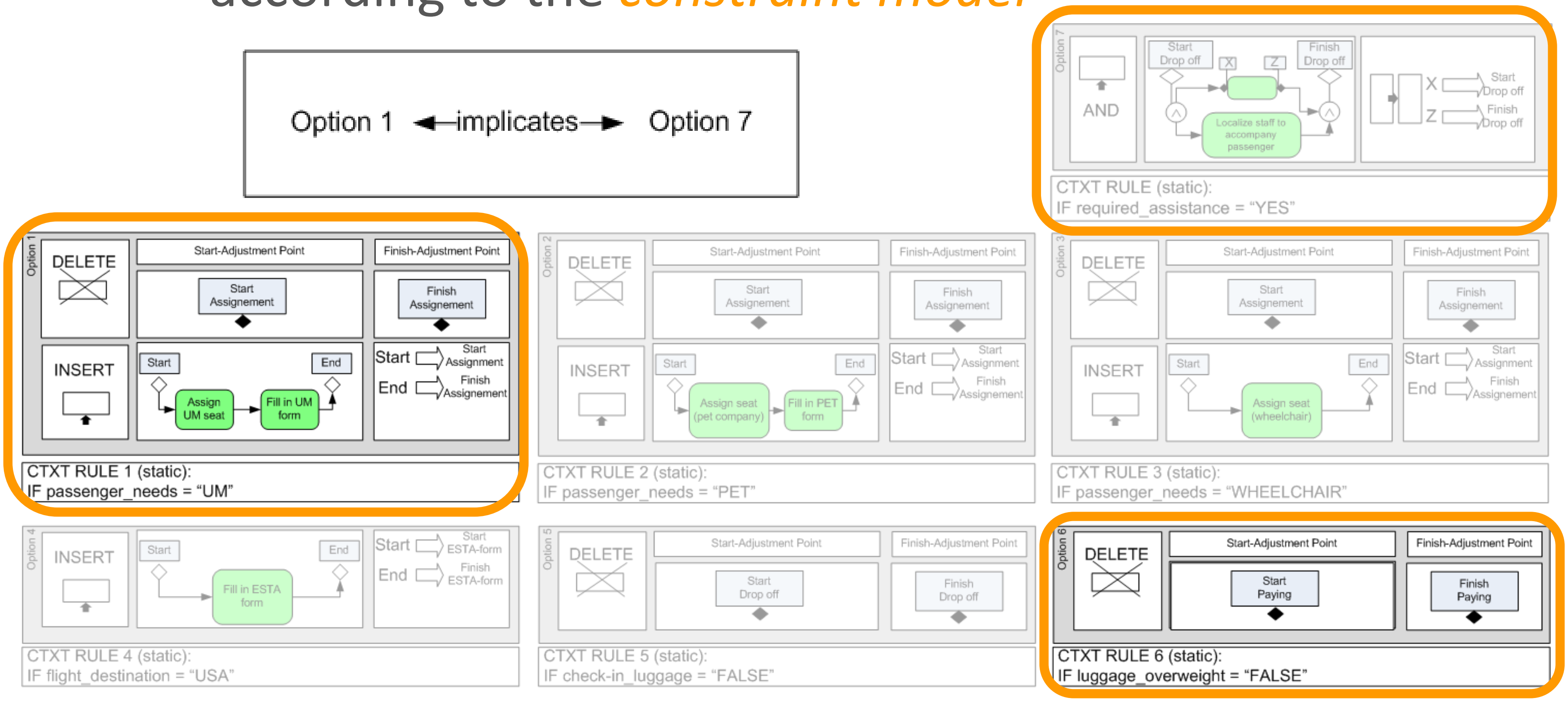
Variant: Unaccompanied minor, check-in luggage, no luggage overweight

- Provop
  2. Select all *change options* satisfying the given context



Variant: Unaccompanied minor, check-in luggage, no luggage overweight

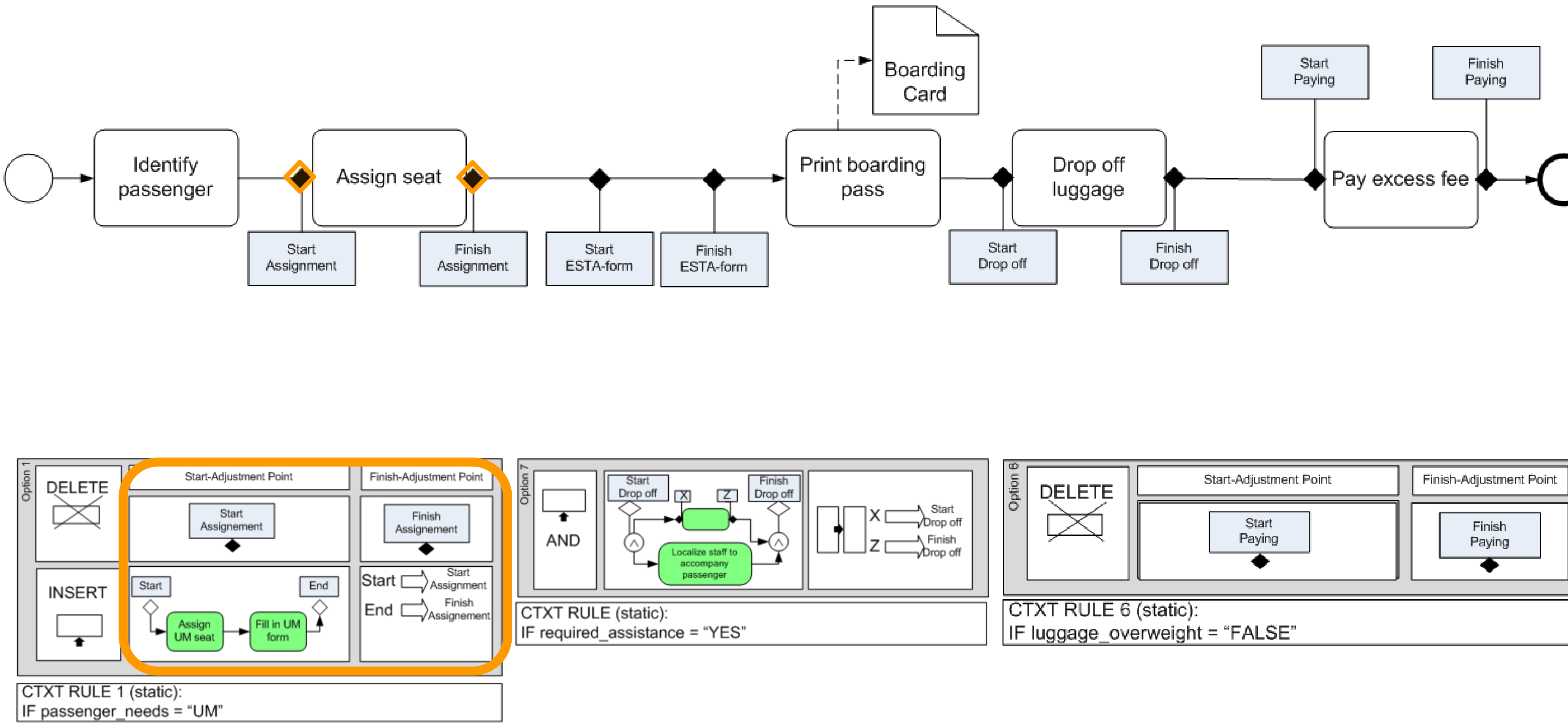
- Provop
  3. Determine whether all options can be applied according to the *constraint model*



Variant: Unaccompanied minor, check-in luggage, no luggage overweight

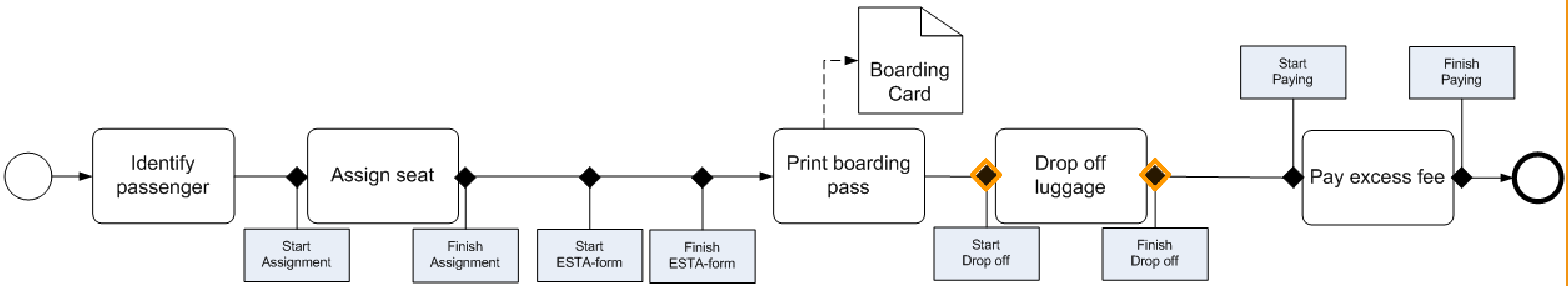
Understandability Task
Process Variant Extraction

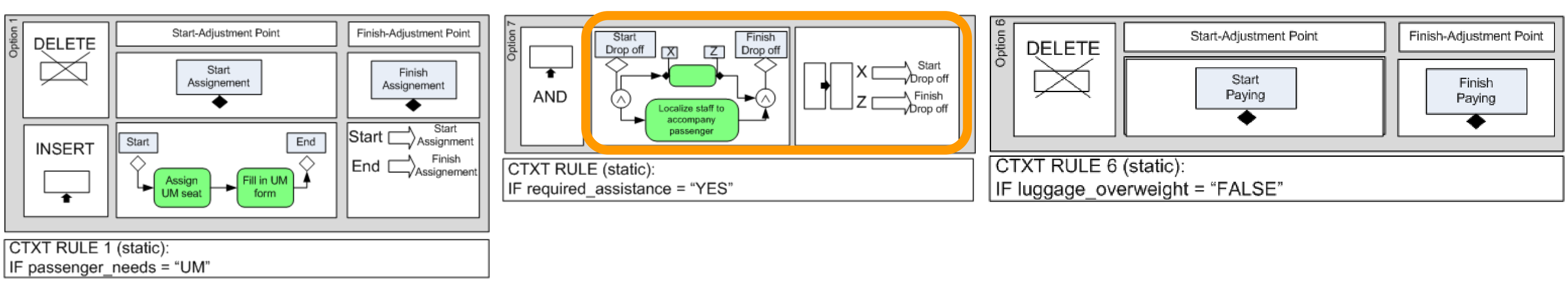
- Provop
  4. Locate the *variation points* where options apply



Variant: Unaccompanied minor, check-in luggage, no luggage overweight

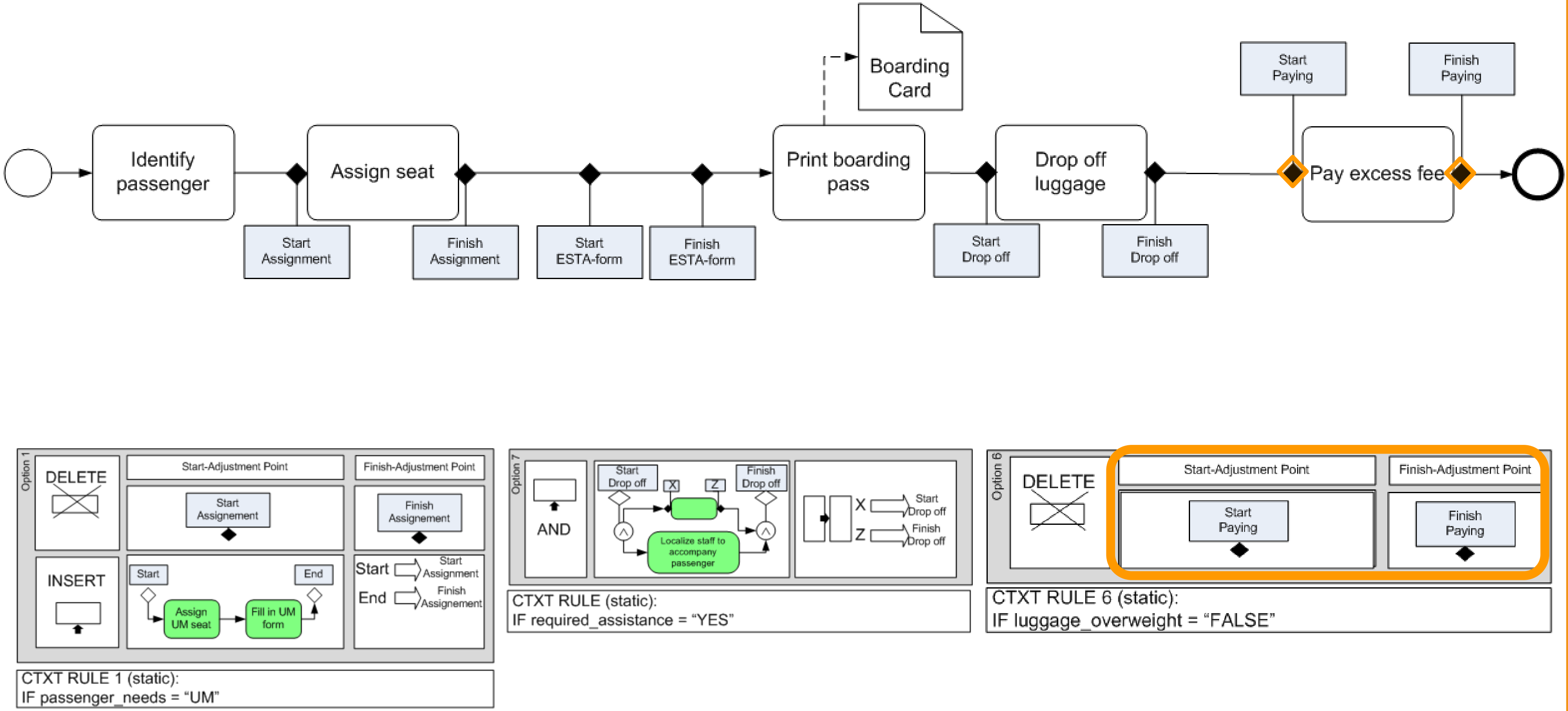
- Provop
  - 4. Locate the *variation points* where options apply





Variant: Unaccompanied minor, check-in luggage, no luggage overweight

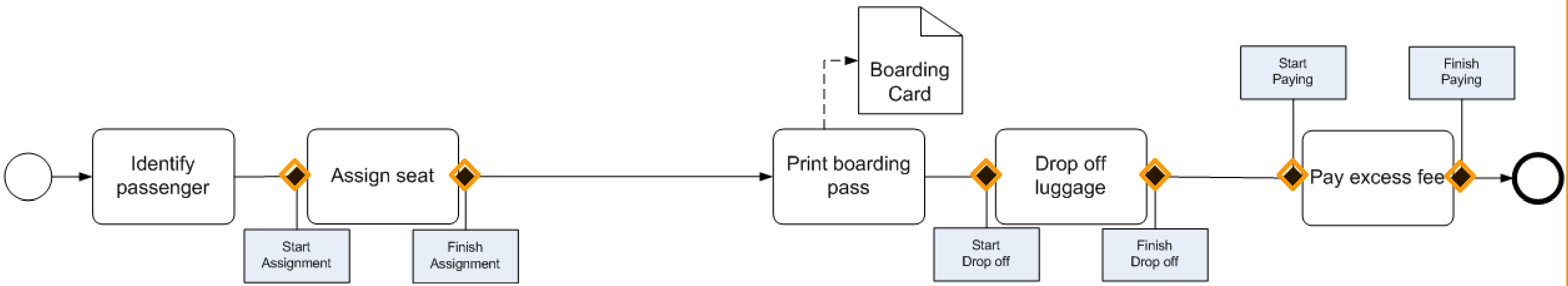
- Provop
  4. Locate the *variation points* where options apply

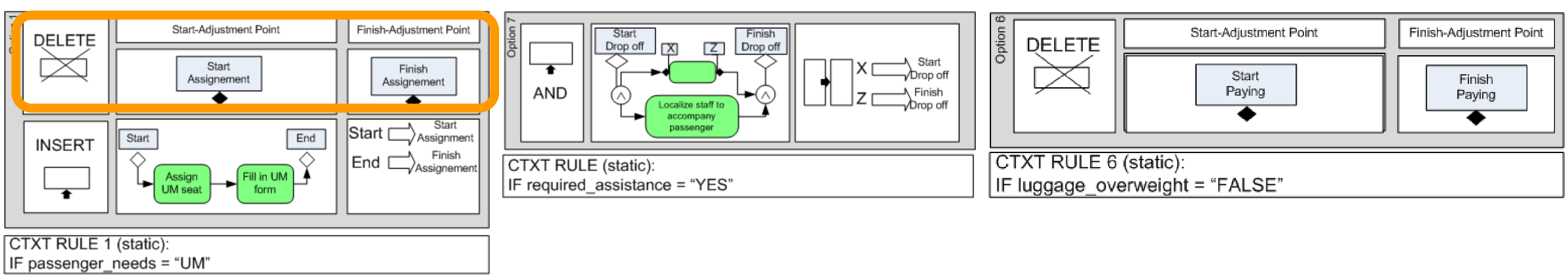


Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

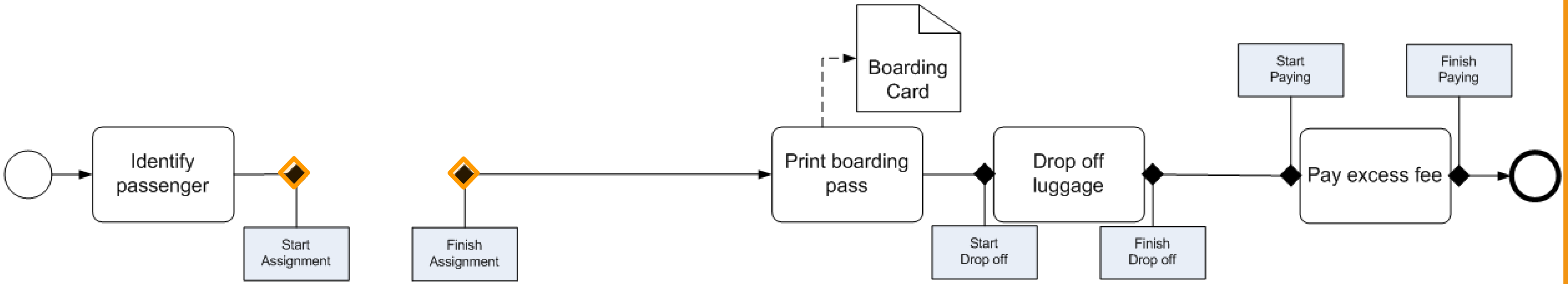
- Provop
  - 4. Mentally integrate *change options* into the *base model*

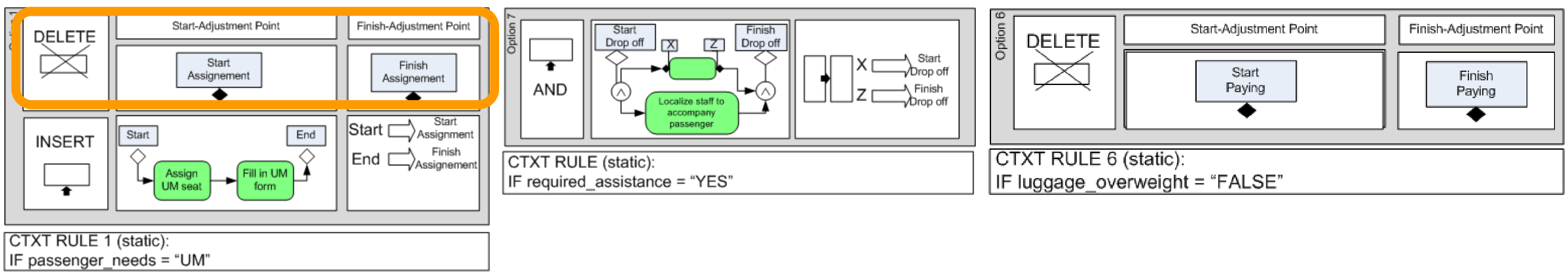




Variant: Unaccompanied minor, check-in luggage, no luggage overweight

- Provop
  4. Mentally integrate *change options* into the *base model*

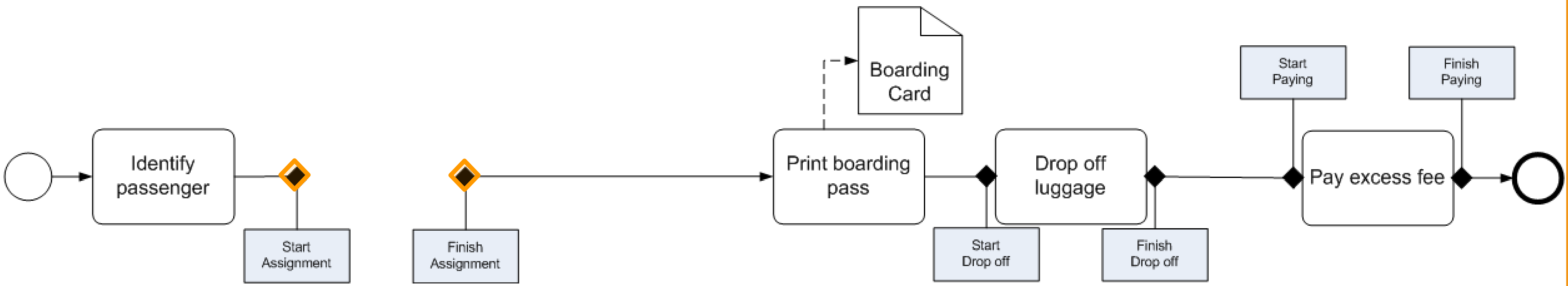


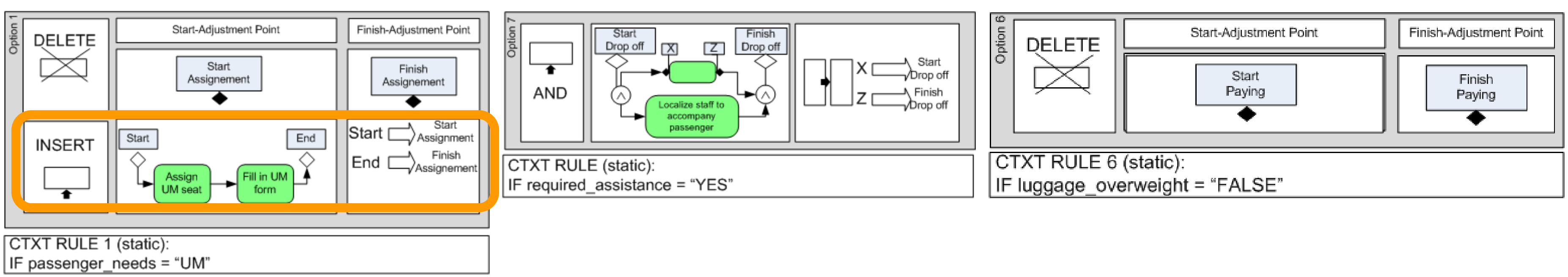


Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

- Provop
  4. Mentally integrate *change options* into the *base model*

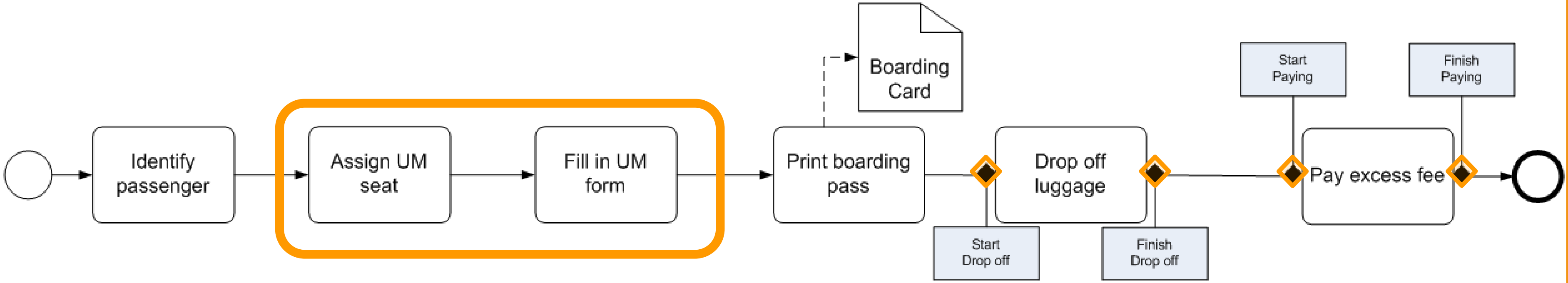


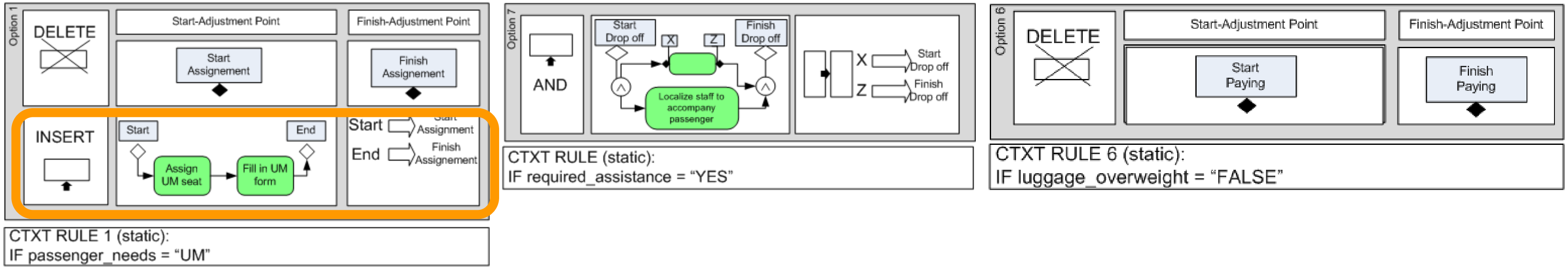


Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

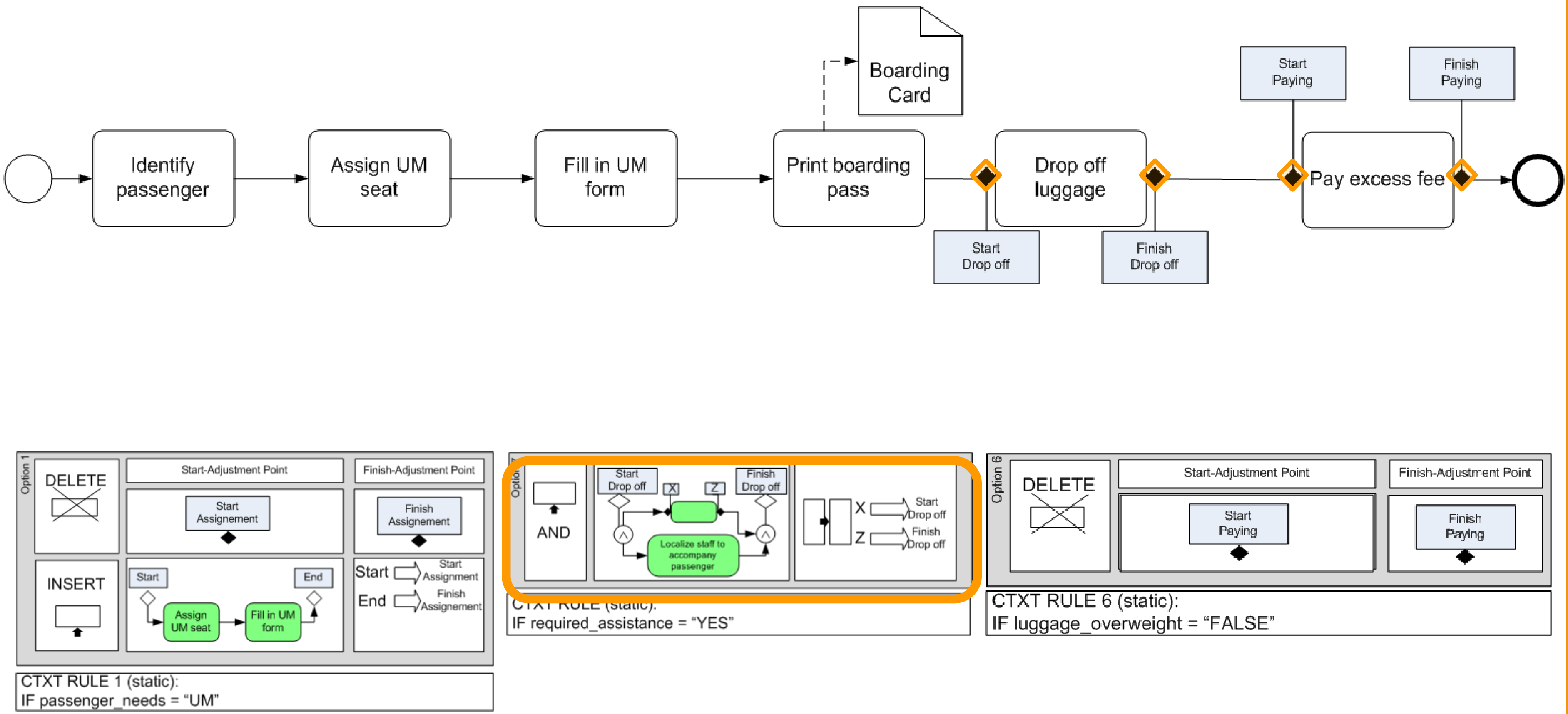
- Provop
  - 4. Mentally integrate *change options* into the *base model*





Variant: Unaccompanied minor, check-in luggage, no luggage overweight

- Provop
  4. Mentally integrate *change options* into the *base model*

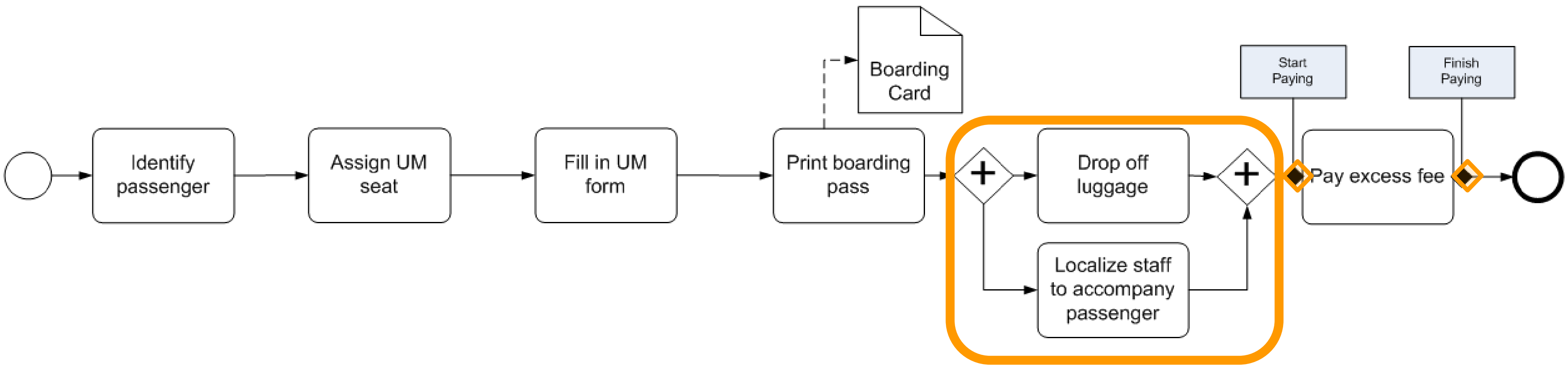


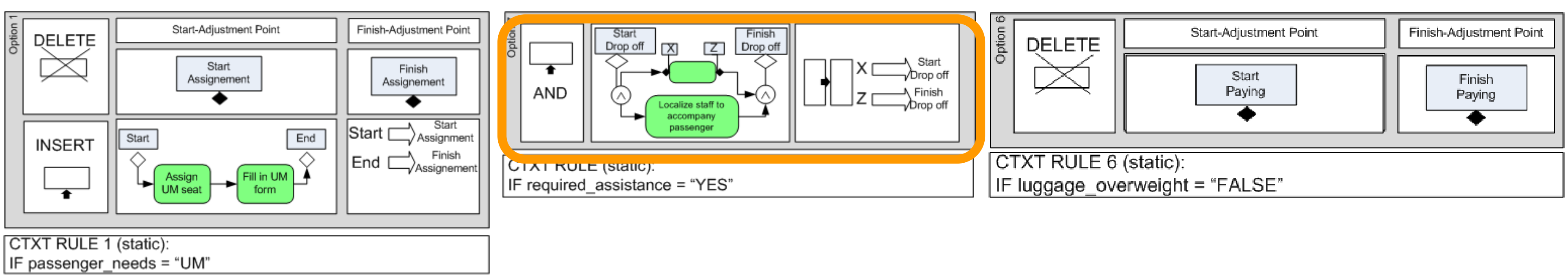
Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

- Provop

  4. Mentally integrate *change options* into the *base model*



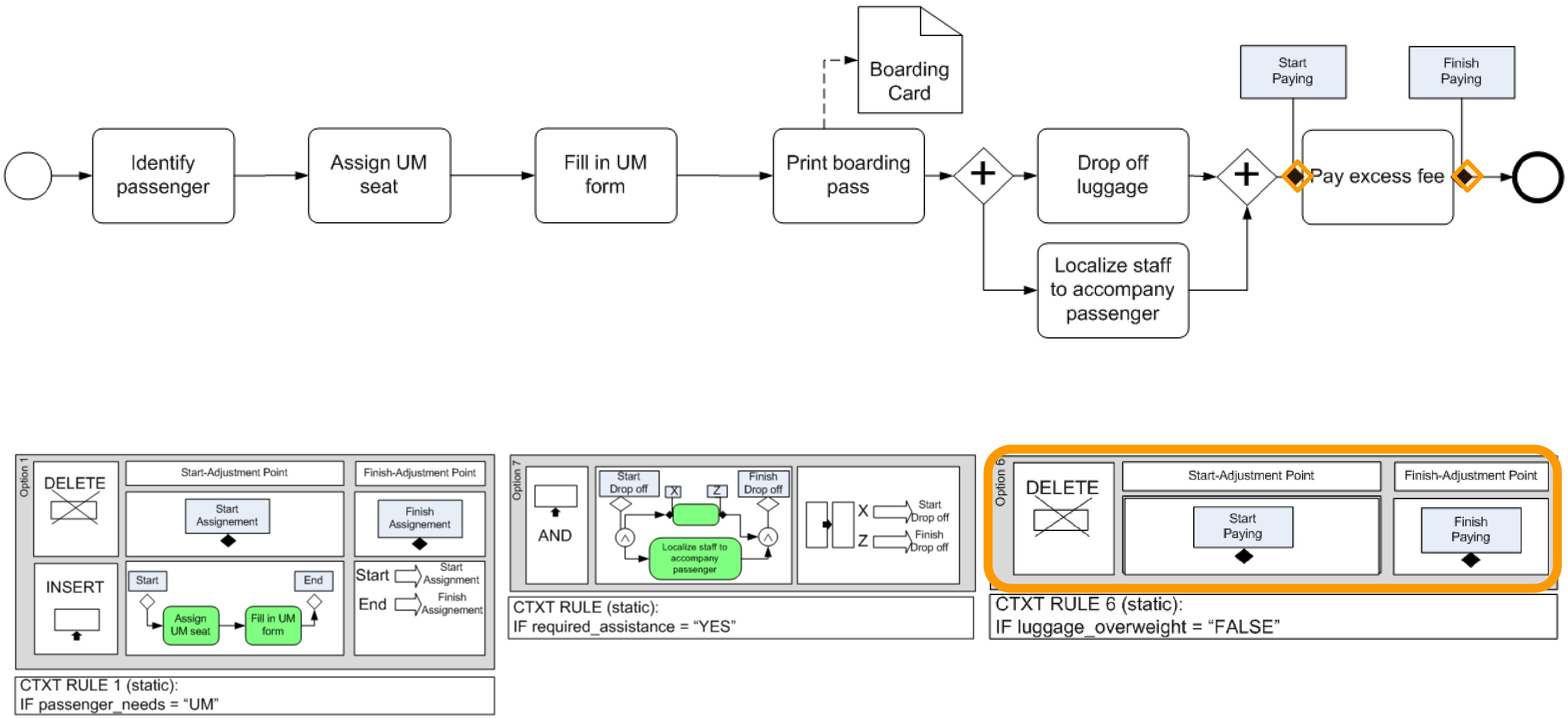


Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

- Provop

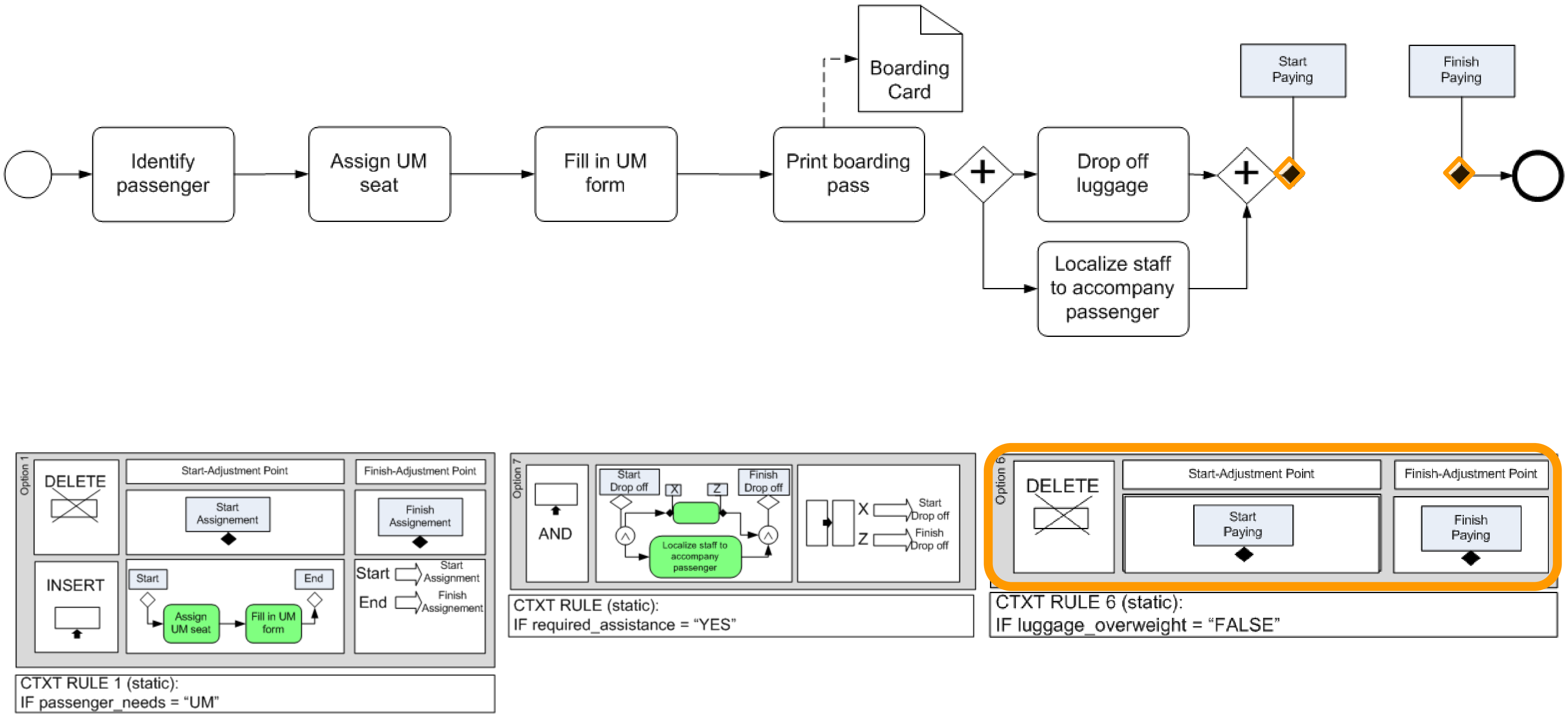
   4. Mentally integrate *change options* into the *base model*



Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

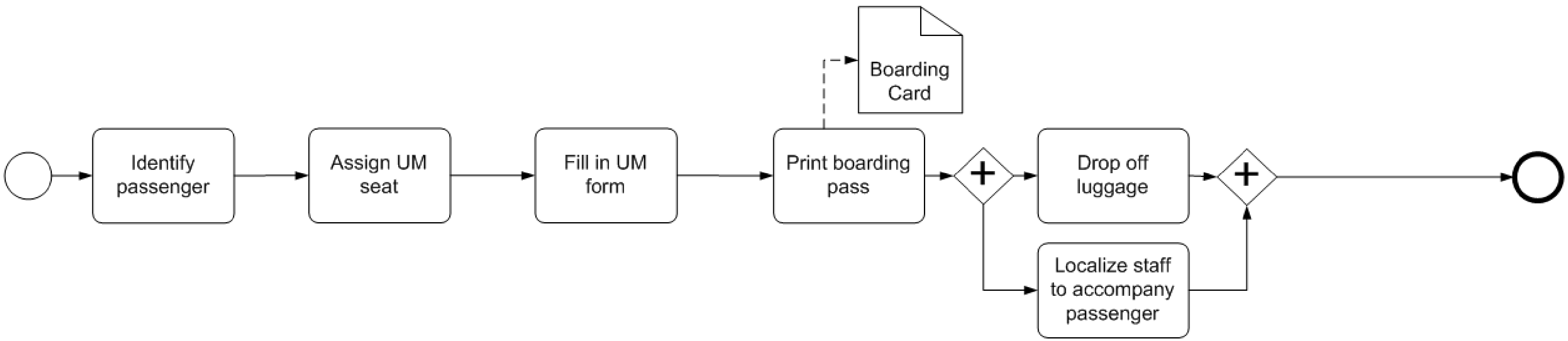
- Provop
  - 4. Mentally integrate *change options* into the *base model*

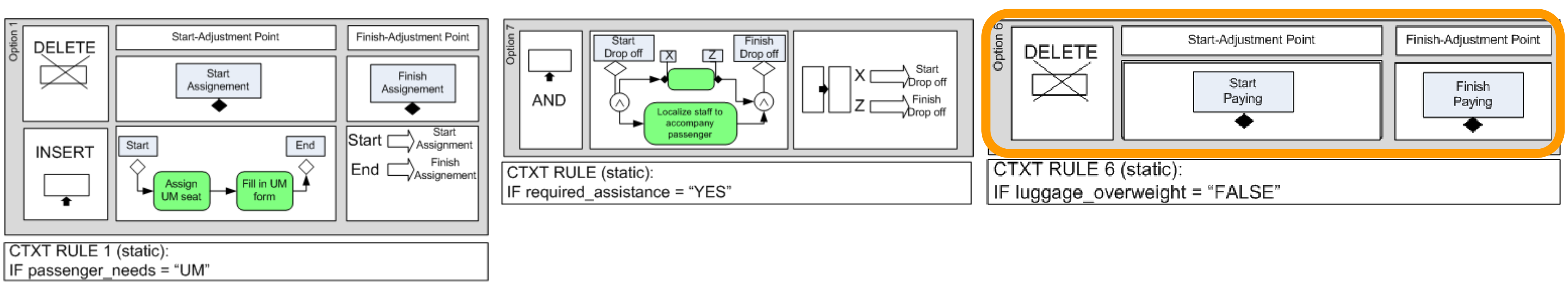


Variant: Unaccompanied minor, check-in luggage, no luggage overweight

Understandability Task
Process Variant Extraction

- Provop
  4. Mentally integrate *change options* into the *base model*





Variant: Unaccompanied minor, check-in luggage, no luggage overweight

- Provop

Mental process followed in Provop:

1. Examine all change options and their Boolean expressions
2. Select all change options satisfying the given context
3. Determine whether all options can be applied according to the constraint model
4. Locate the variation points where options apply
5. Mentally integrate change options into the base model

- Provop
  - Cognitive discussion
    - Two main operations:
      - Selecting change options
        - » Boolean expression need to be evaluated
        - » All are expressed in terms of context variables
        - » Check for conflicts in the constraint model
      - Applying change options into the base model
        - » Determine by the change distance between the base model and the variant to be derived.
        - » The type of operation influences the complexity (delete vs. insert)

Language Support Comparison
Structural vs Behavioural

	C-EPC	Provop
Variation Point	+	+
Alternative process elements	F, B, O, I	F, B
Alternative process element context	+	+
Alternative process element relationships	+/-	+
Variation point resolution time	-	-

functional (F), behavioral (B), organizational (O), Informational (I) perspectives

- Discussion
  - Modelling elements
    - In C-EPC are mainly deleted
    - In Provop can be added, deleted, and moved
    - » Deletion presumably involves less cognitive effort

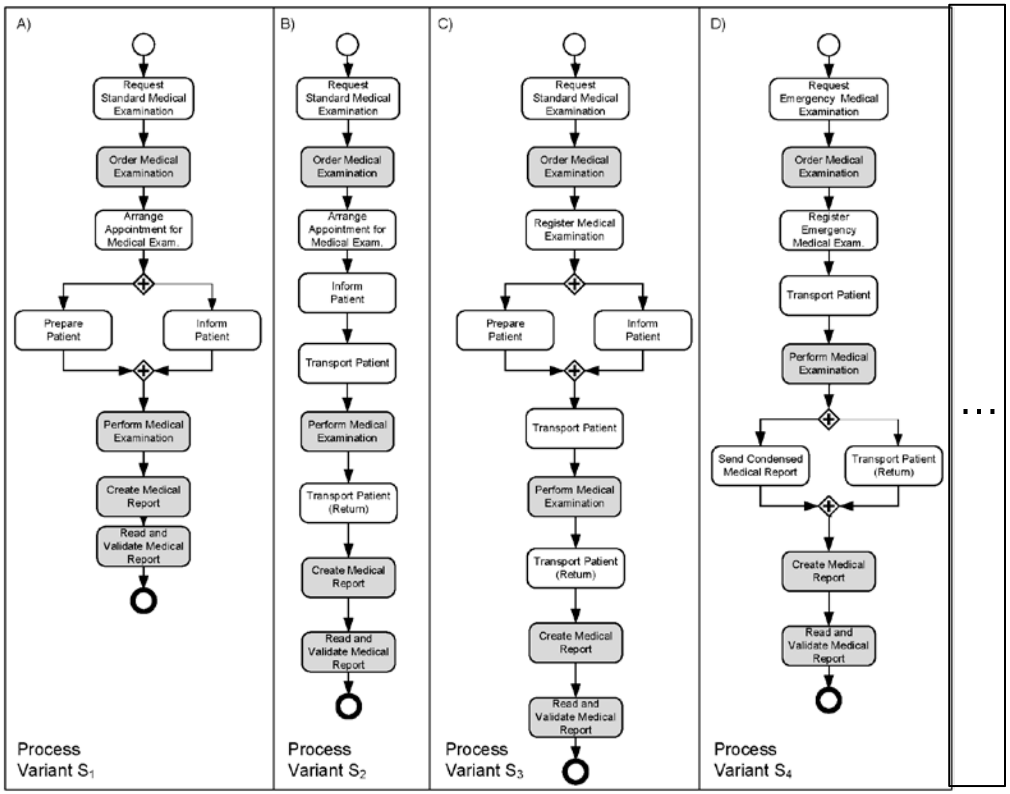
Qualitative Comparison

- Discussion
  - Variability modelling elements
    - Single artefact in C-EPC
      - Configurable process model
        - » Works for small models
        - » Overload for large models
    - Separate artefacts in Provop
      - base model, change options, options constraint, context model
        - » Abstraction mechanisms favour model understanding

Qualitative Comparison

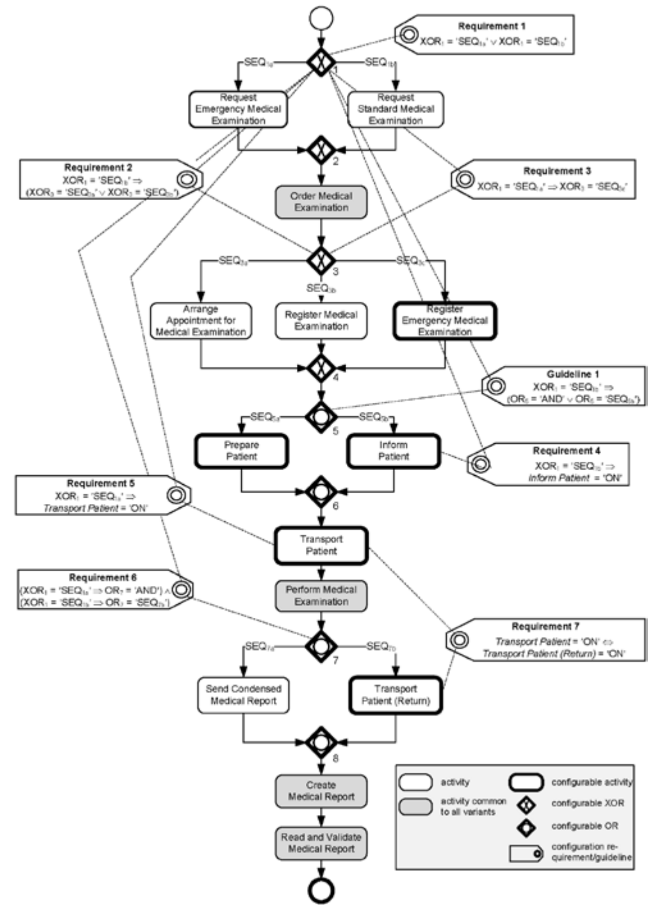
- Discussion
  - Boolean expressions
    - C-EPC: Represented in terms of the model structure
      - Forces to keep track of previous decisions
      - Presumably imposing bigger mental effort
    - Provop: Represented in terms of context variables
      - Semantics is explicit in the model
      - » Presumably imposing less mental effort

Business Process Repositories
Merging Process Models: Behavioral Approach



Process (Variant) Models

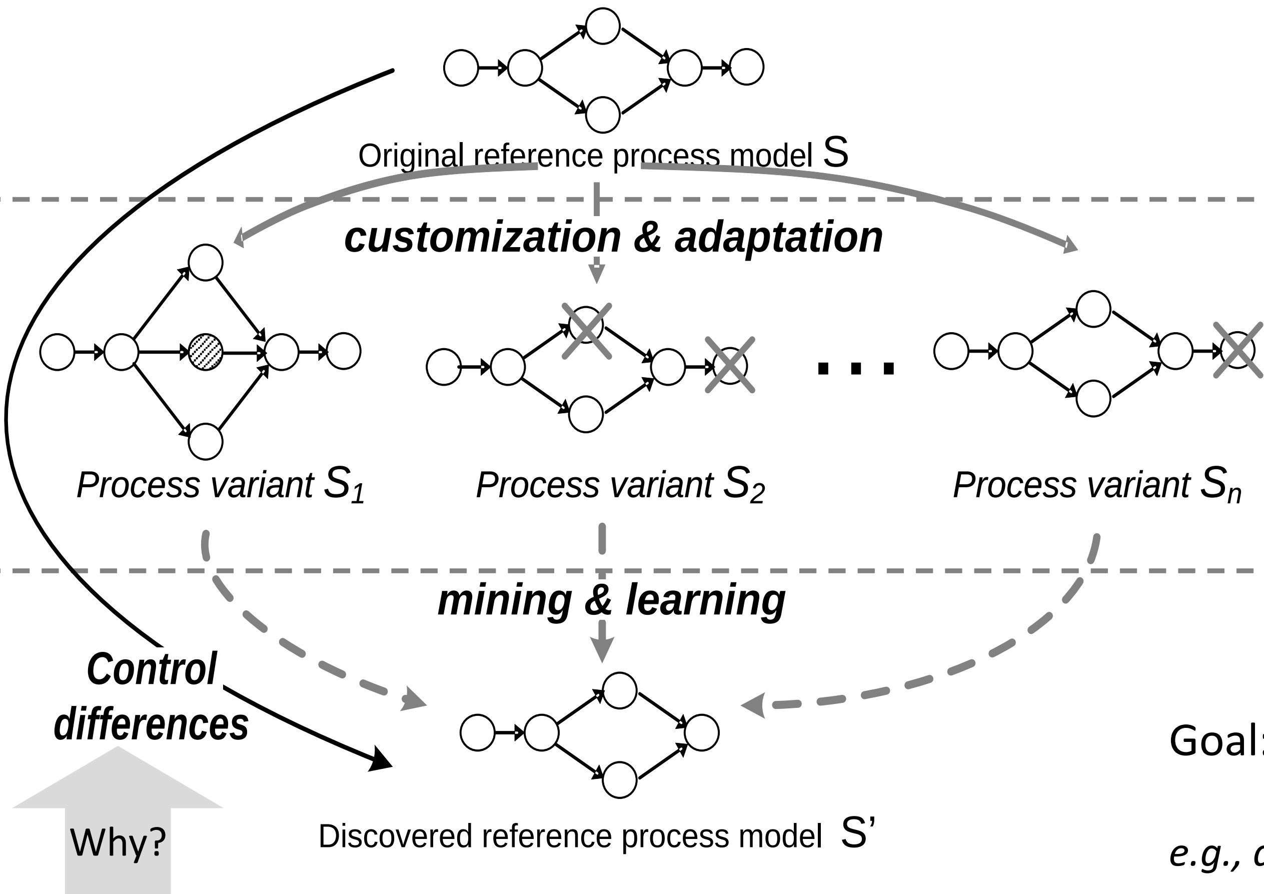
merge



(Configurable) Process Model

La Rosa et al, 2010

Business Process Repositories
Merging Process Models: Structural Approach



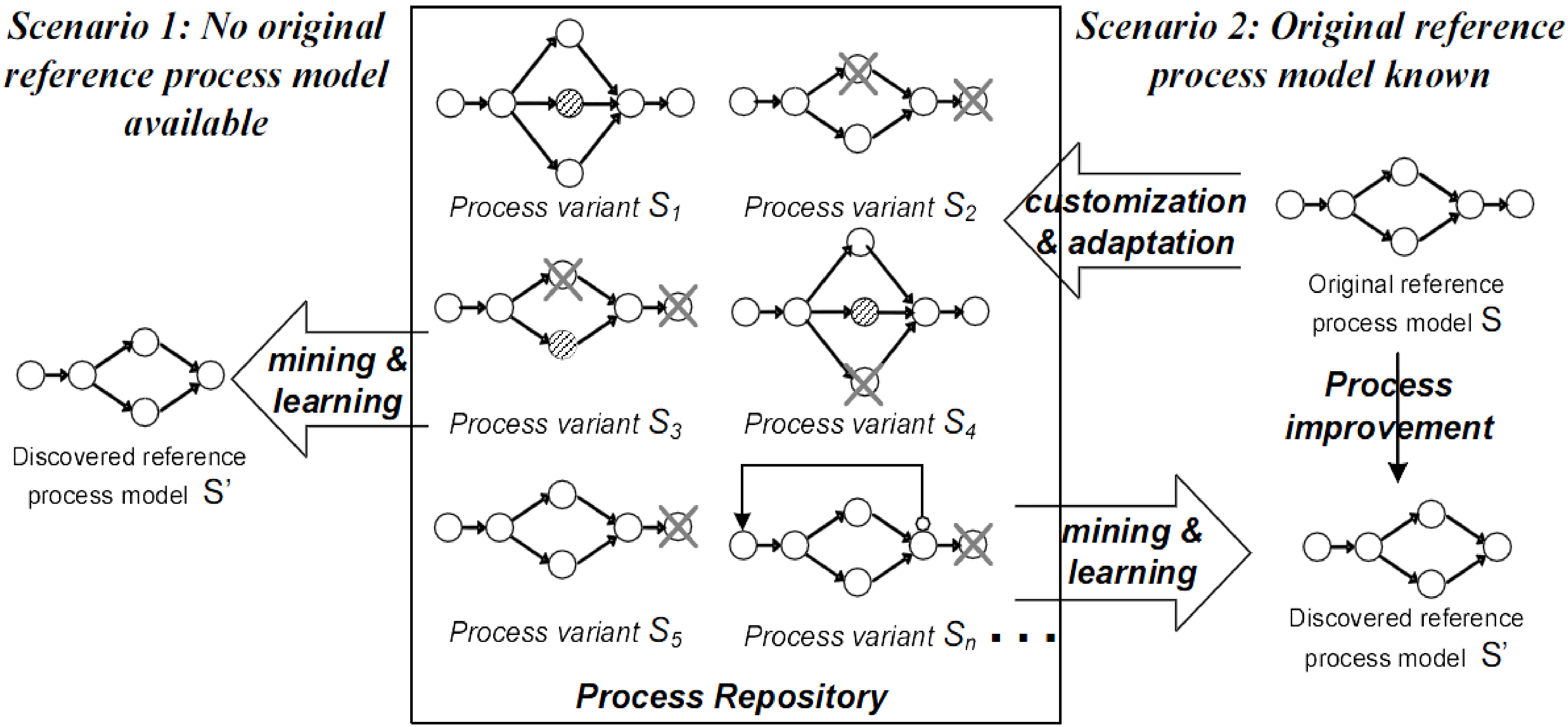
1. Limit the efforts to update reference model
2. Avoid spagatti-like structure
3. Obtain the flexibility to only perform the important changes

1. Configurations are very costly
2. Variants are difficult to maintain

Goal: Less configurations!

*e.g., discover a new reference process model having lower **average weighted distance** to the variants than the original reference process model has!*

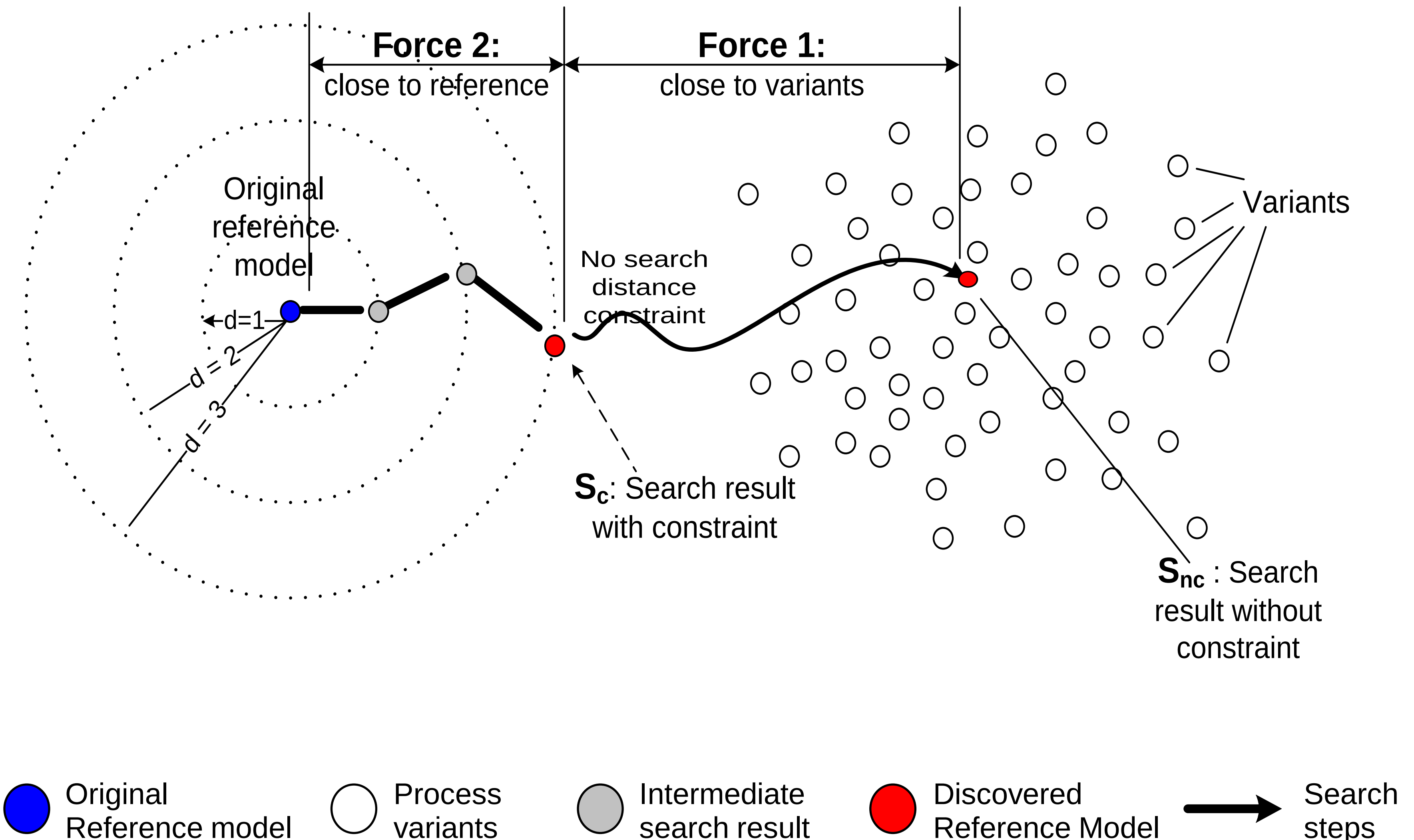
Business Process Repositories
Merging Process Models: Structural Approach



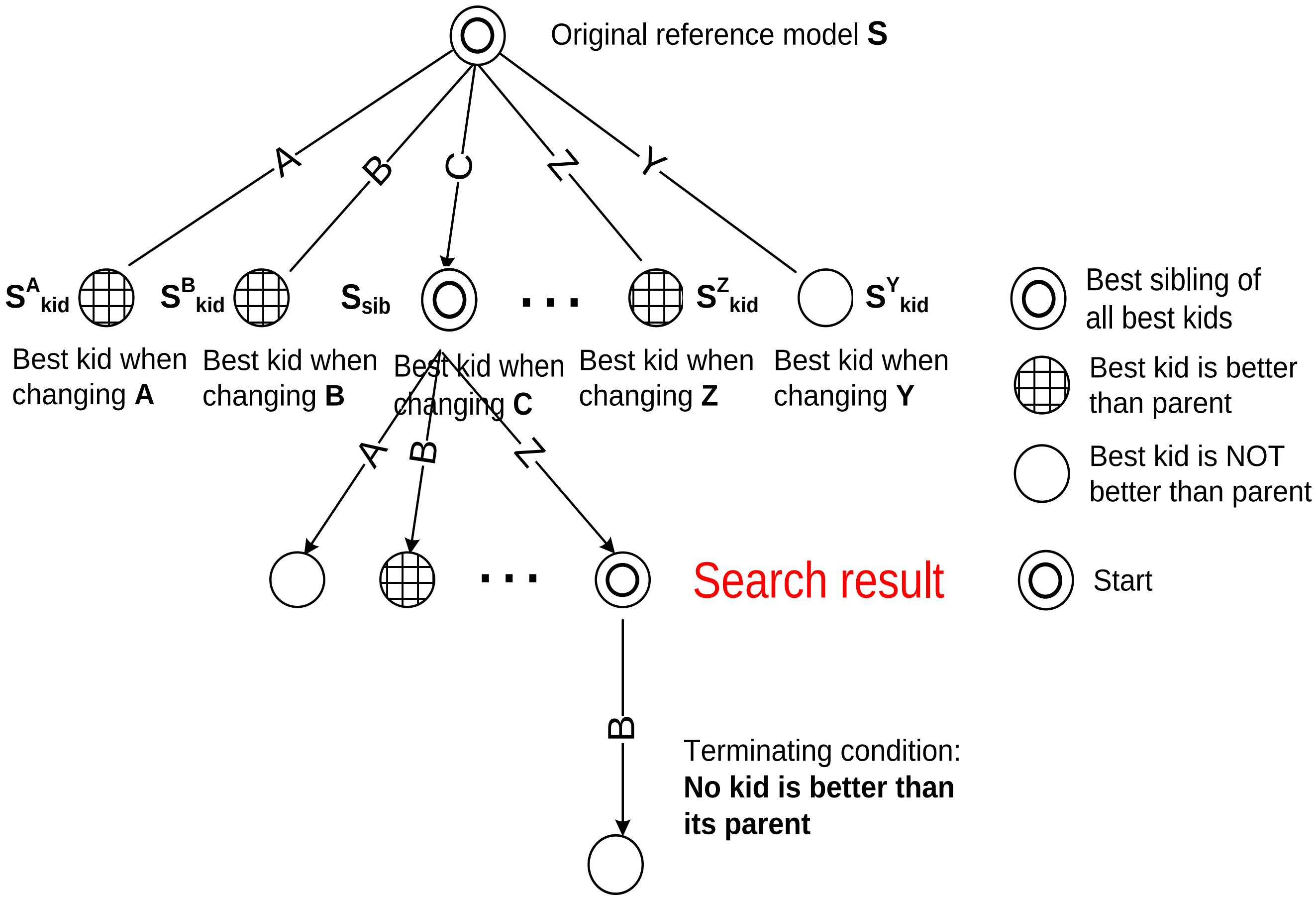
Goal: *Discover a (new) reference process model which requires less configuration efforts*

Li, Reichert, & Wombacher 2012

Applying Heuristics Search to Scenario 2

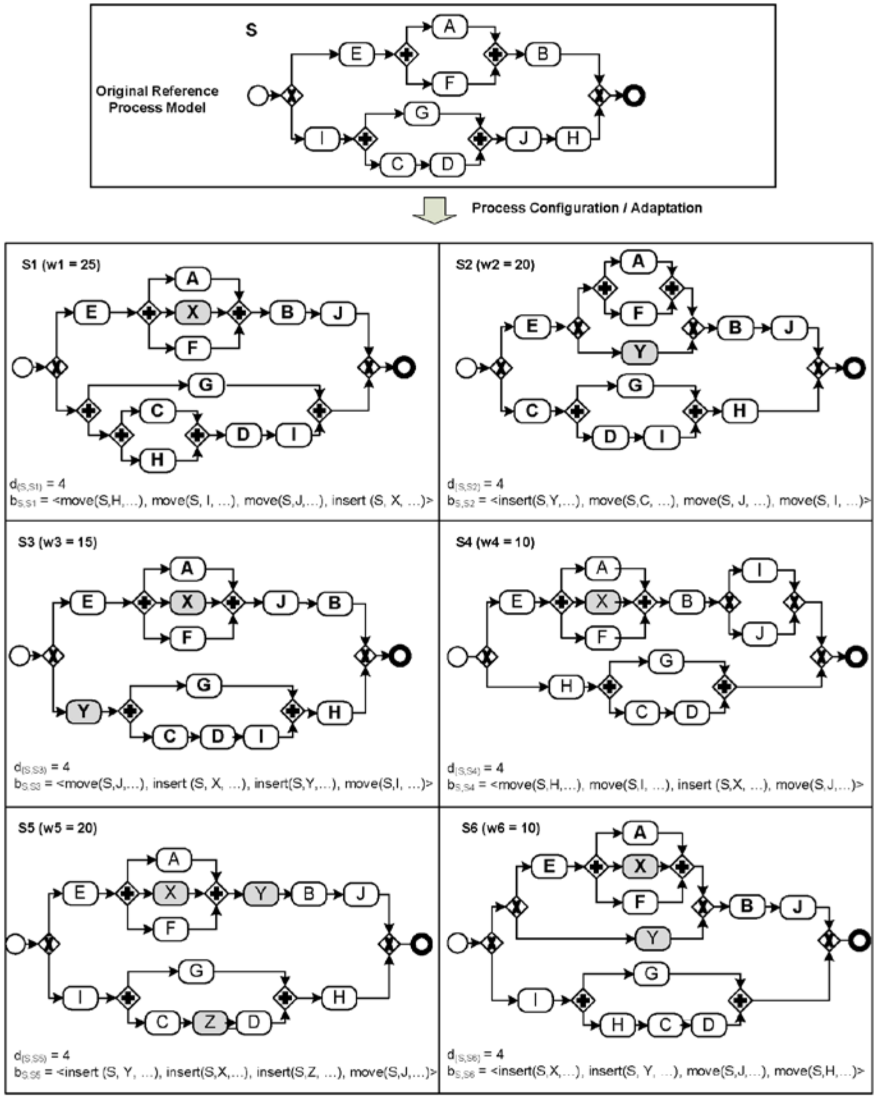


Applying Heuristics Search to Scenario 2



Search tree based on best kids S_{kid}^{aj}

Merging Process Models: Structural Approach



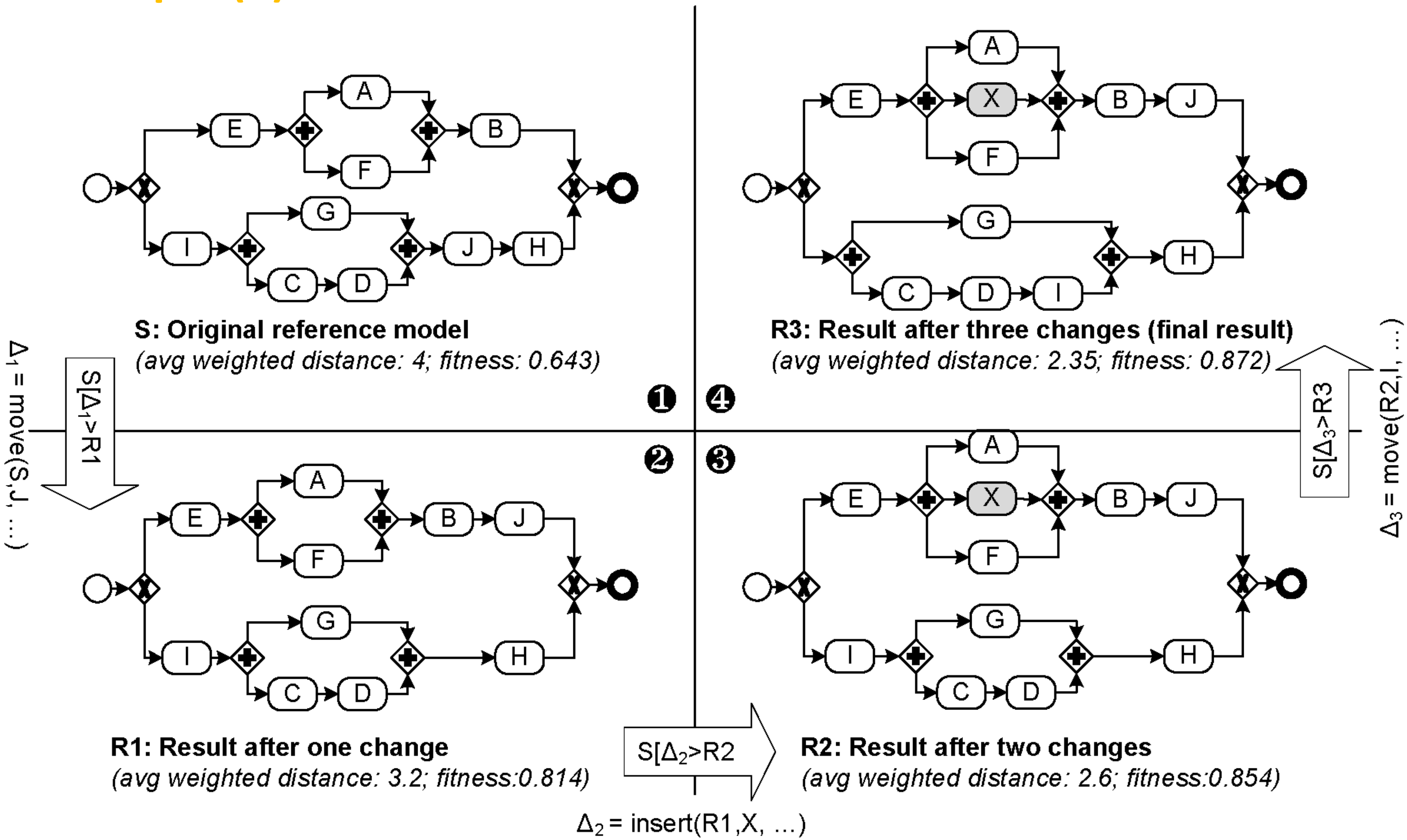
Example (1)

Average (weighted) distance of S to the variants: 4 high-level changes (NP-hard complexity)

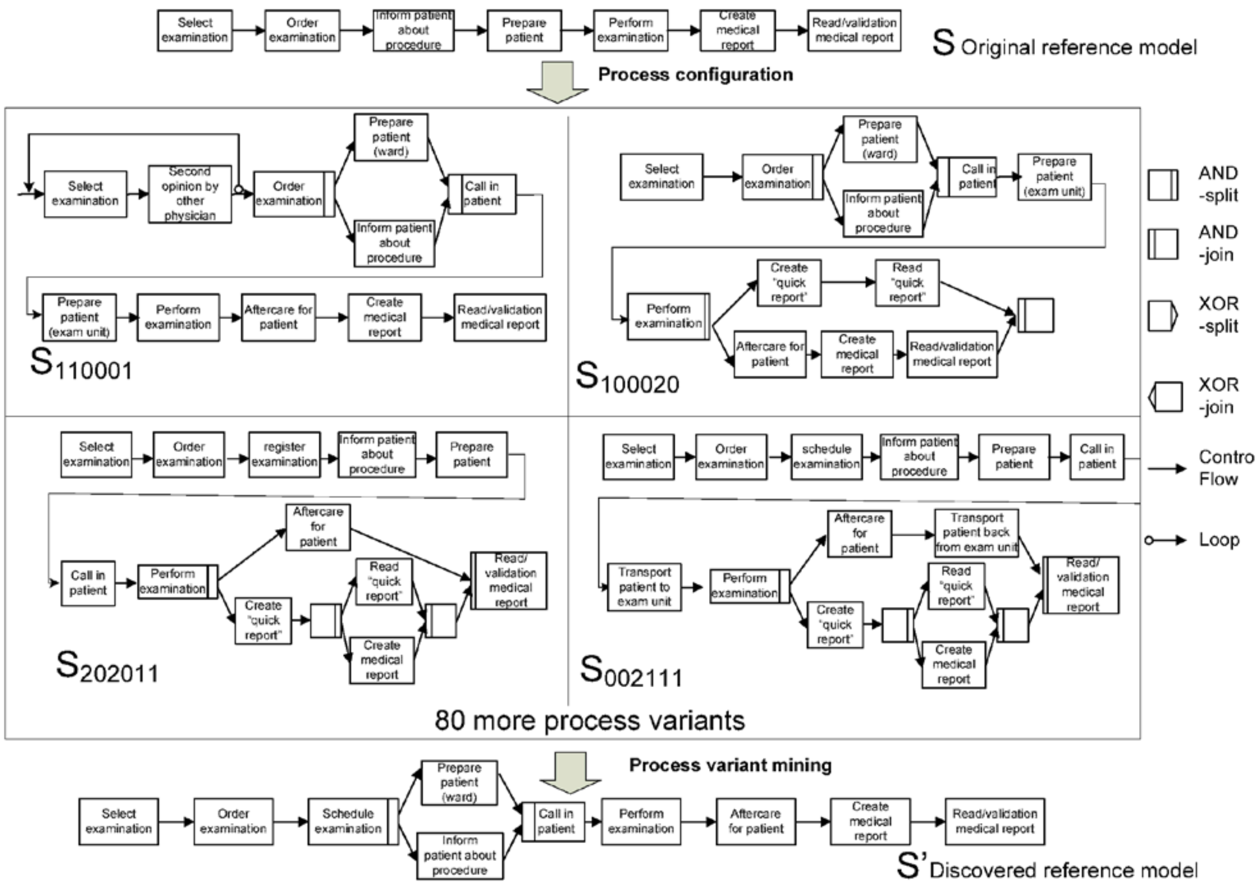
Can we find a model closer to the variants by performing saying only 3 changes of the old reference model?

Merging Process Models: Structural Approach

Example (2)



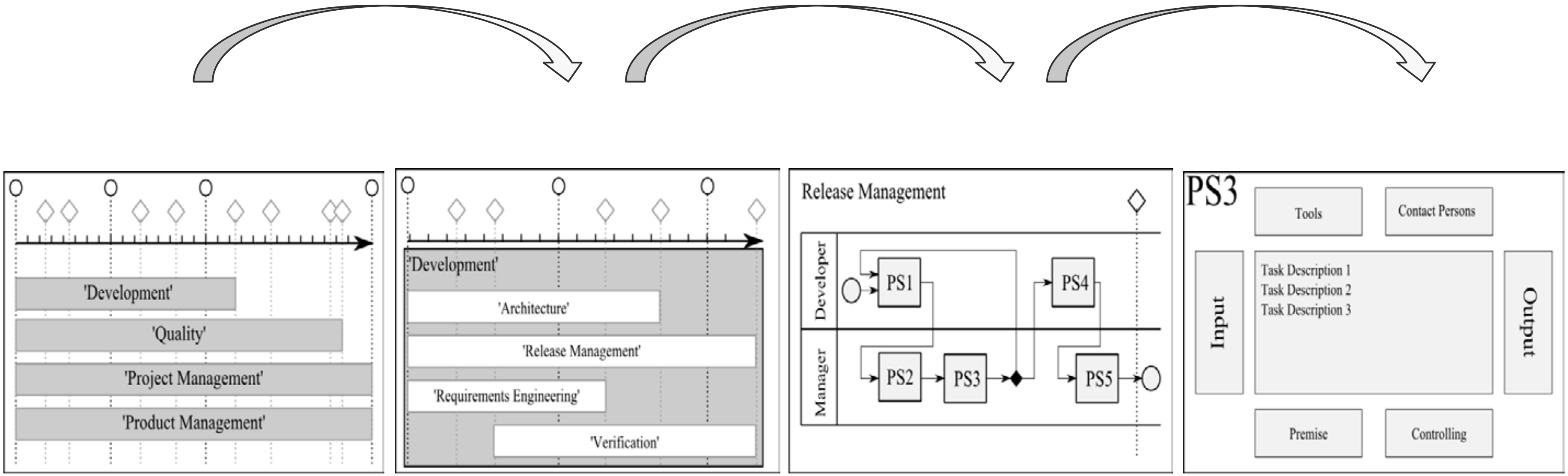
Healthcare Case Study



Business Process Repositories

Navigating in Respositories and Large Process Models

Practical demands: Navigate in Large Process Models



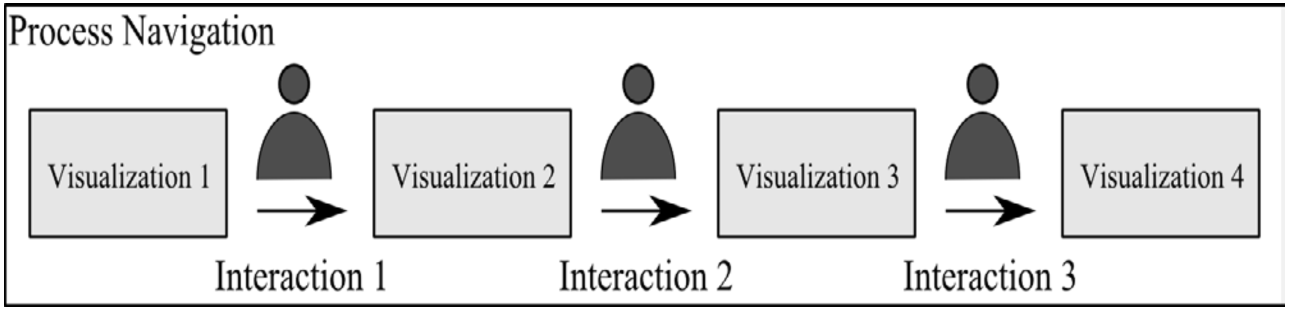
Process World | Process Areas | Processes | Process Step

el of granularity

- Navigation within one single dimension (forward/backward)
- Views are predefined
- No comprehensive tool support available

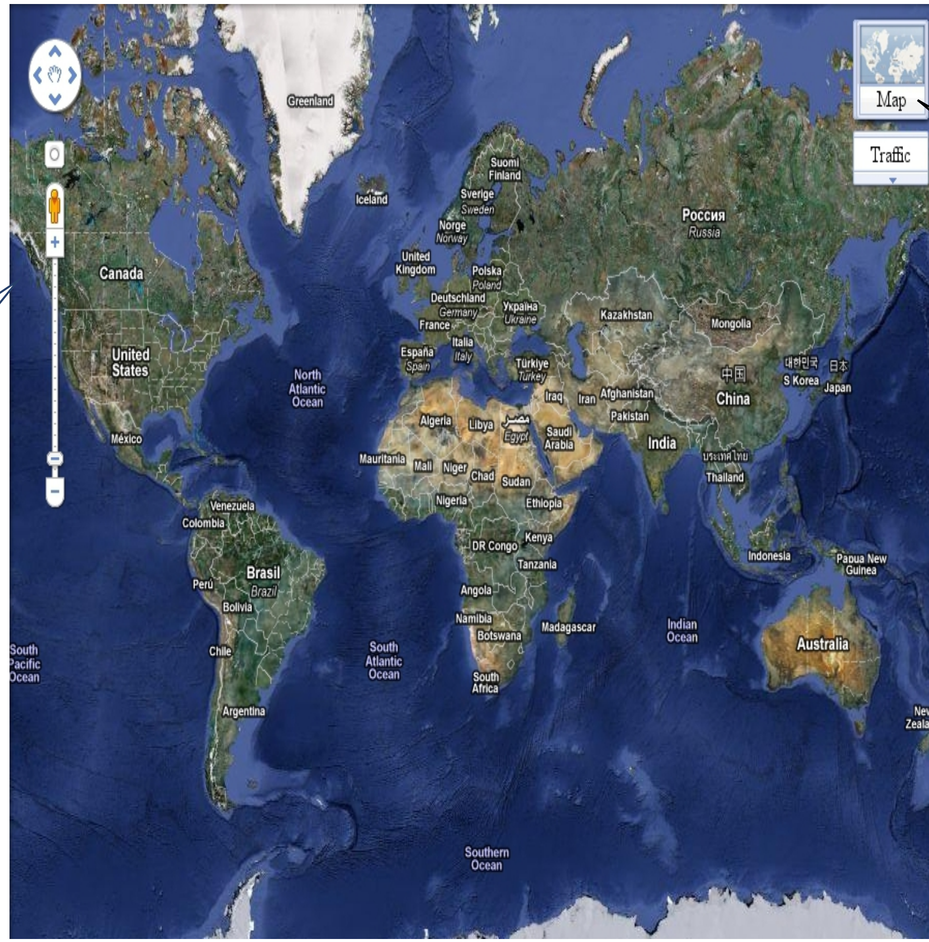
Process Navigation (niPRO)

- comprises a sequence of user interactions
- allows process participants to navigate from a default visualization of a large process to more specific ones



niPRO - Hipp, Mutschler & Reichert, 2012

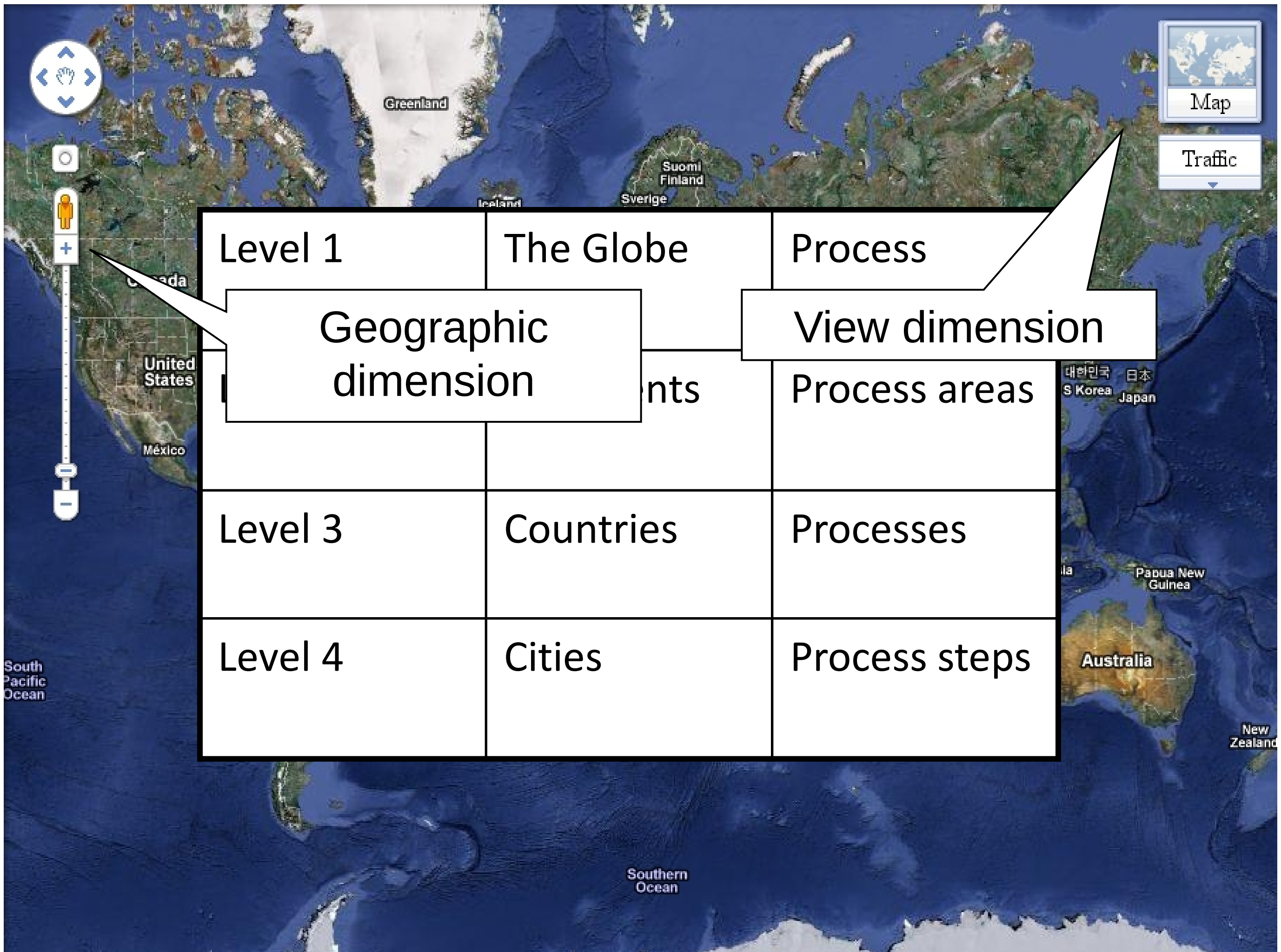
niPRO Core Navigation Model – Inspired by Google Earth



Geographic dimension

View dimension

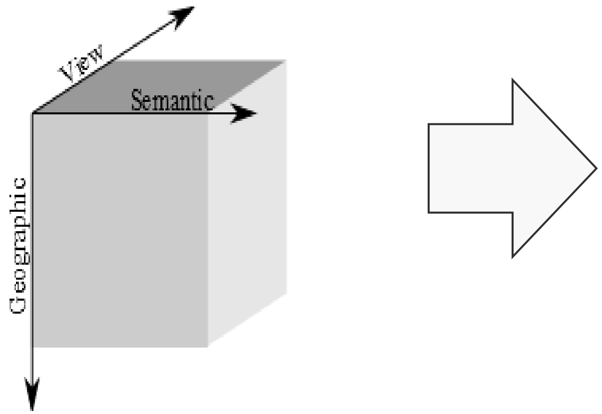
niPRO Core Navigation Model – Inspired by Google Earth

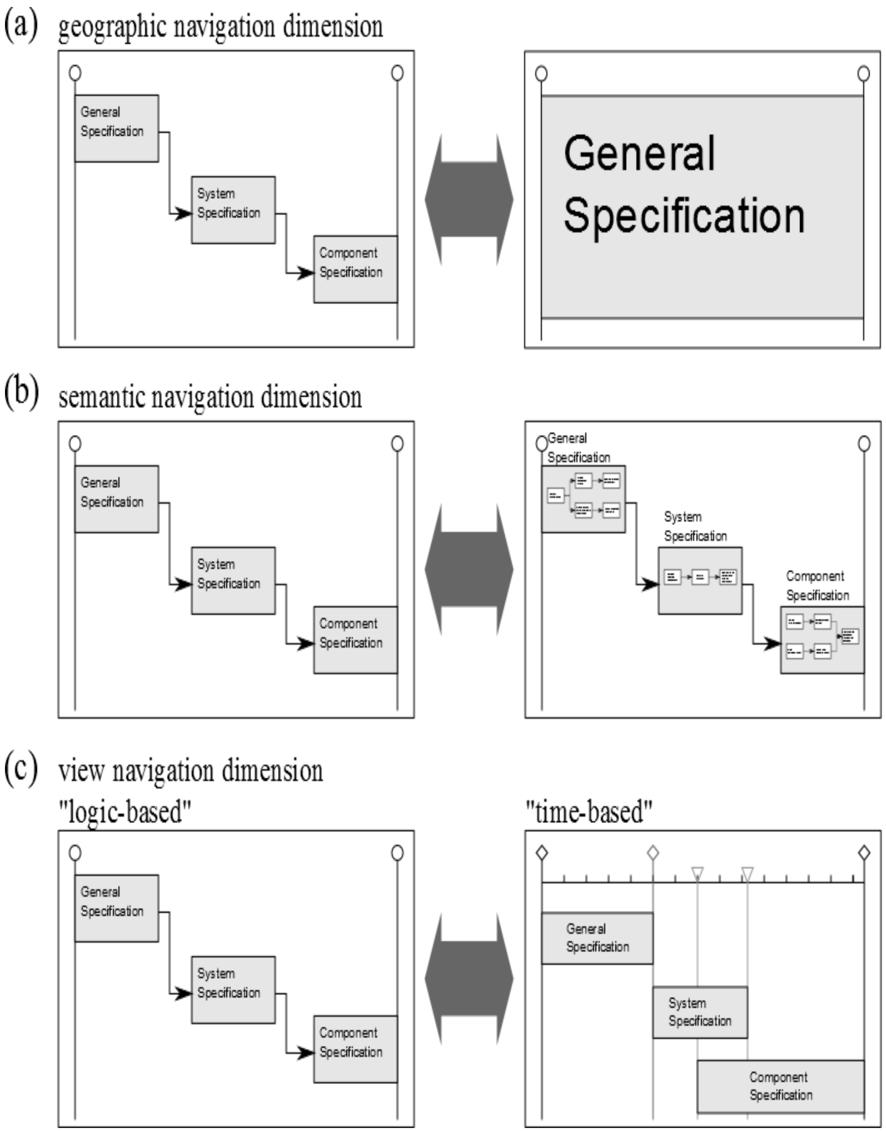


Level 1	The Globe	Process
	nts	Process areas
Level 3	Countries	Processes
Level 4	Cities	Process steps

Business Process Repositories
Navigating in Respositories and Large Process Models

3-dimensional navigation concept



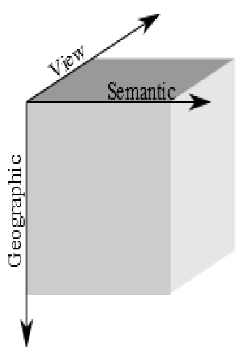


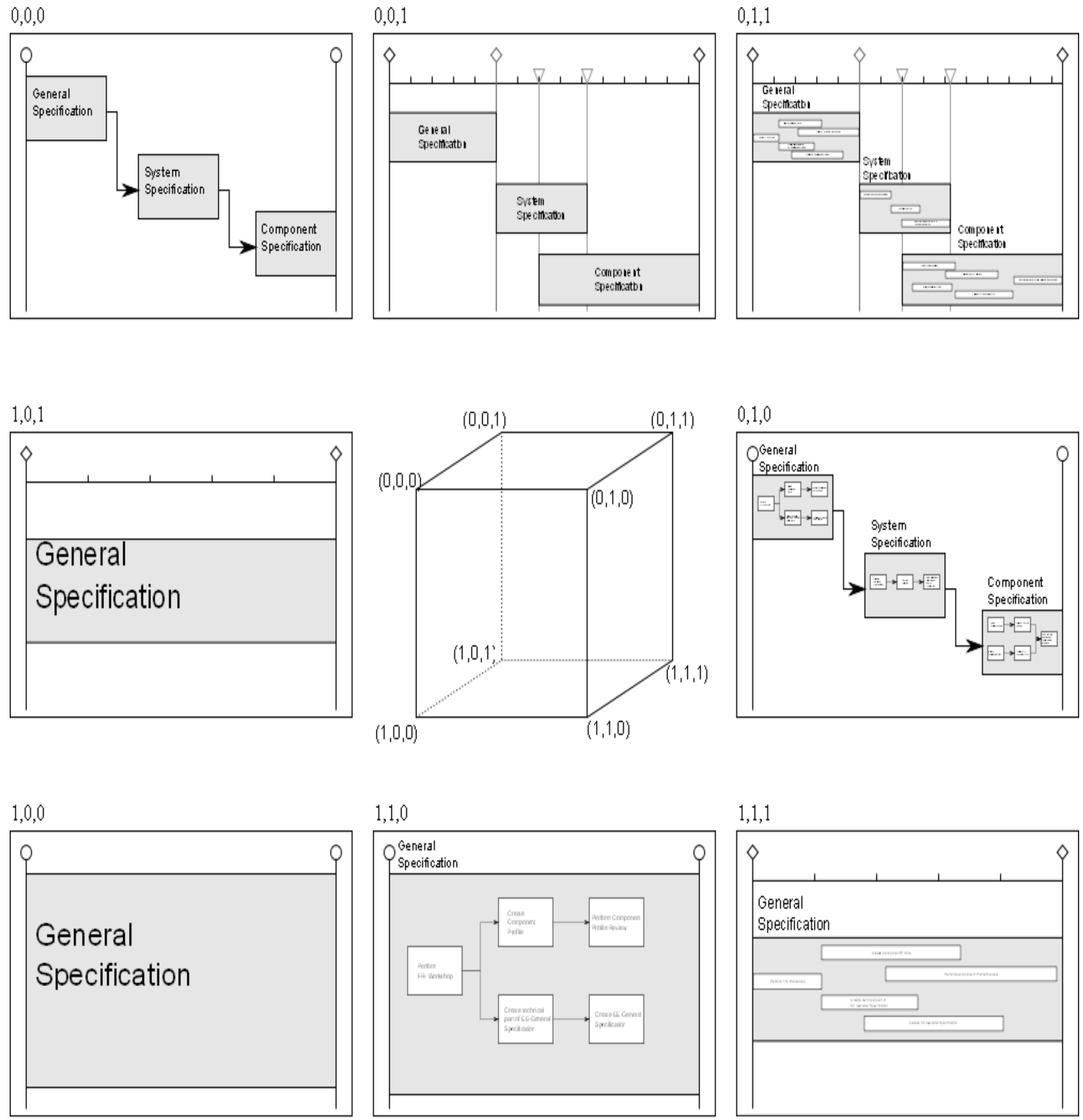
niPRO - Hipp, Mutschler & Reichert, 2012

Business Process Repositories
Navigating in Respositories and Large Process Models

Initial situation

- 3-dimensional navigation space
- Different navigation states (g, s, v)
- State transitions (= user interactions)





Business Process Repositories
Navigating in Respositories and Large Process Models

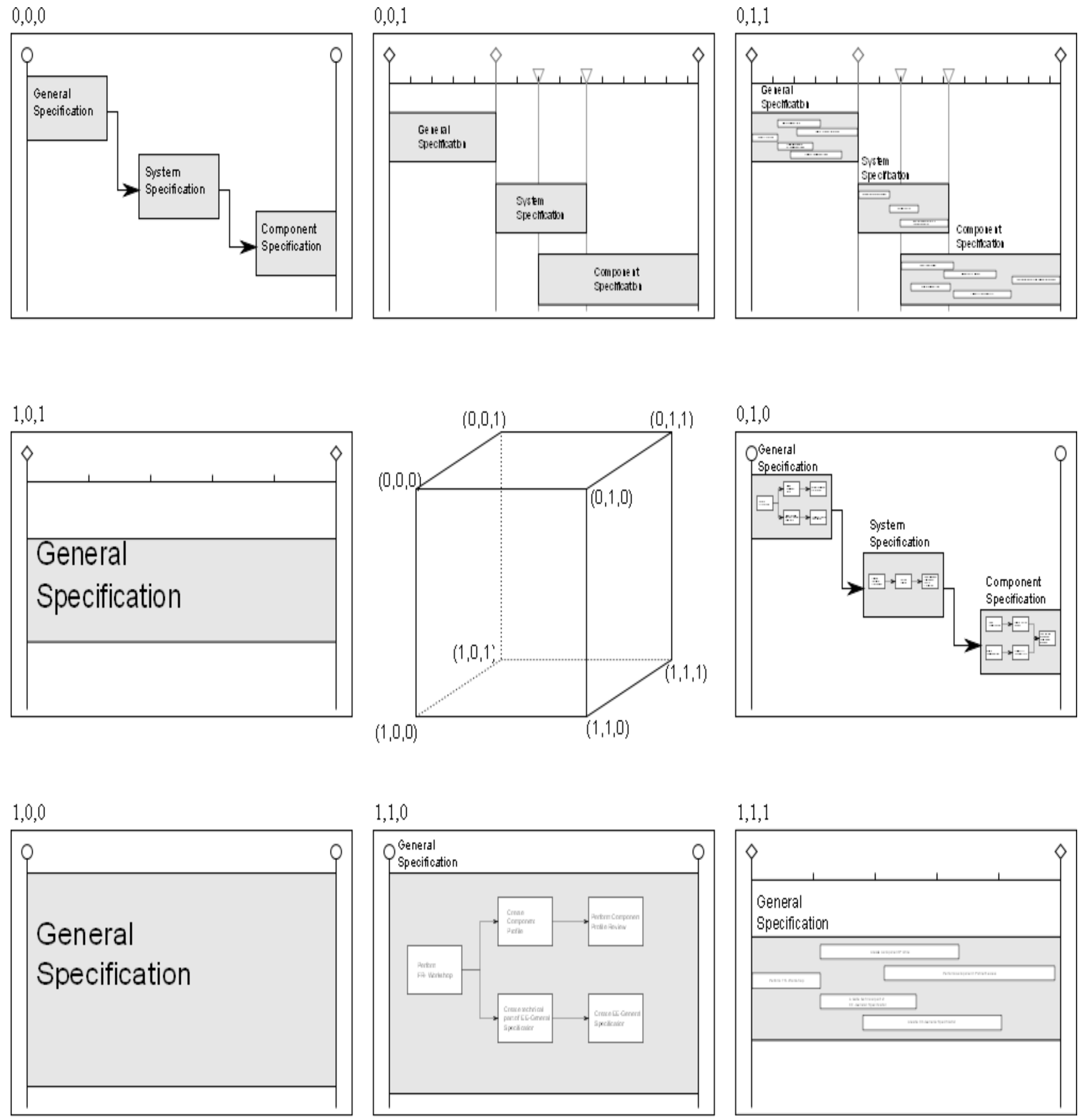
- Some Challenges
  - How to guide the user through the navigation space?
  - How to recommend certain paths within the navigation space (i.e., to reduce number of interactions)?
  - How to remove specific navigation states being not reasonable…
  - How to assist users when the navigation space gets more complex…

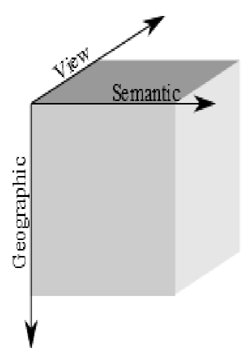
⇒ A *process navigation model* is needed

Business Process Repositories
Navigating in Respositories and Large Process Models

- **Typical use case**

A developer wants to see which process step has to be done, after he completed the current process step.





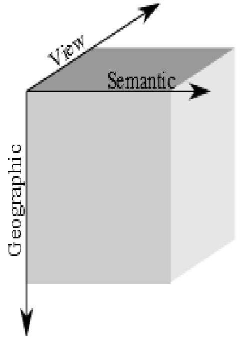
Business Process Repositories
Navigating in Respositories and Large Process Models

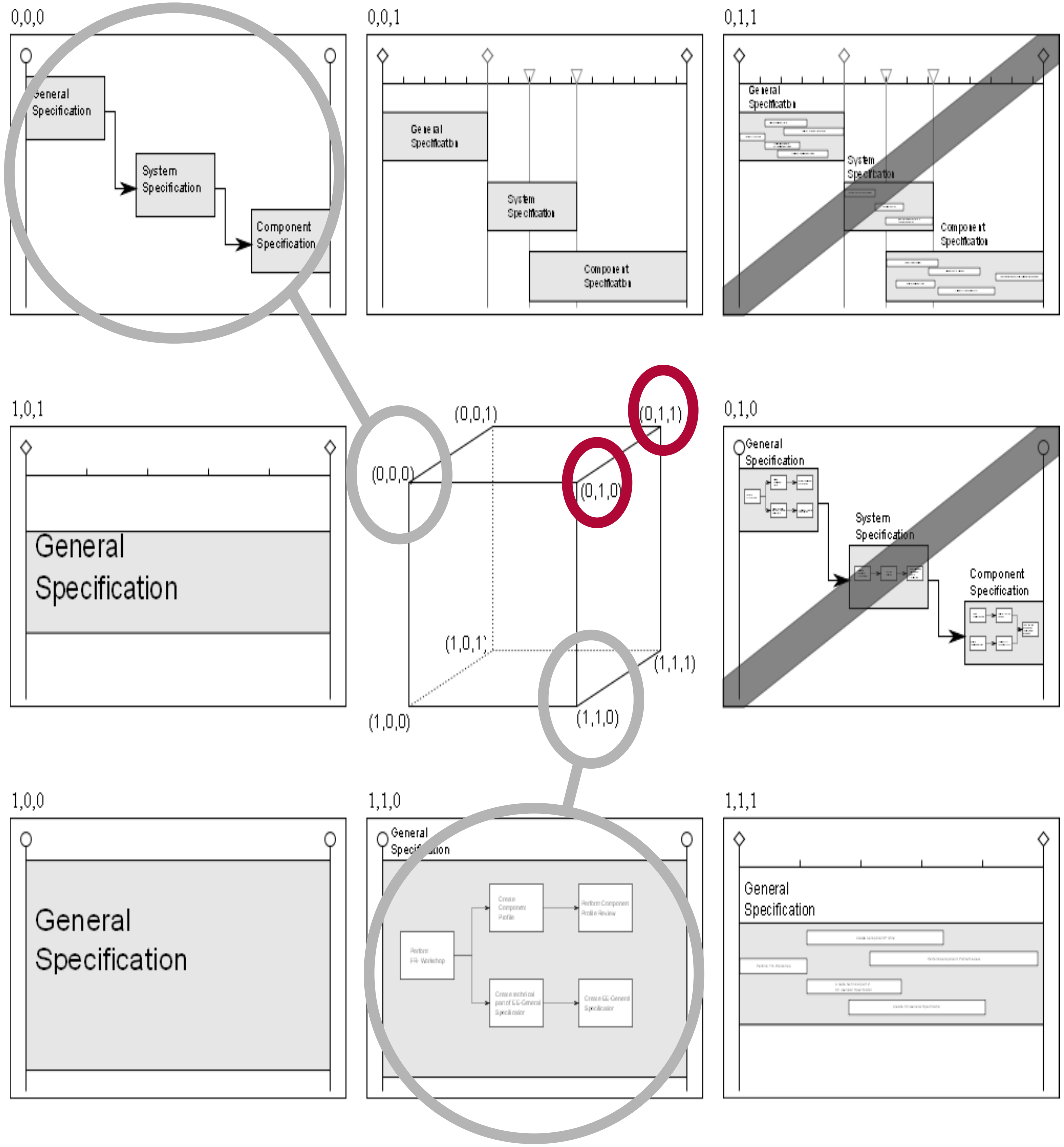
- **Typical use case**

A developer wants to see which process step has to be done, after he completed the current process step.

Start: (0,0,0)

End: (1,1,0)

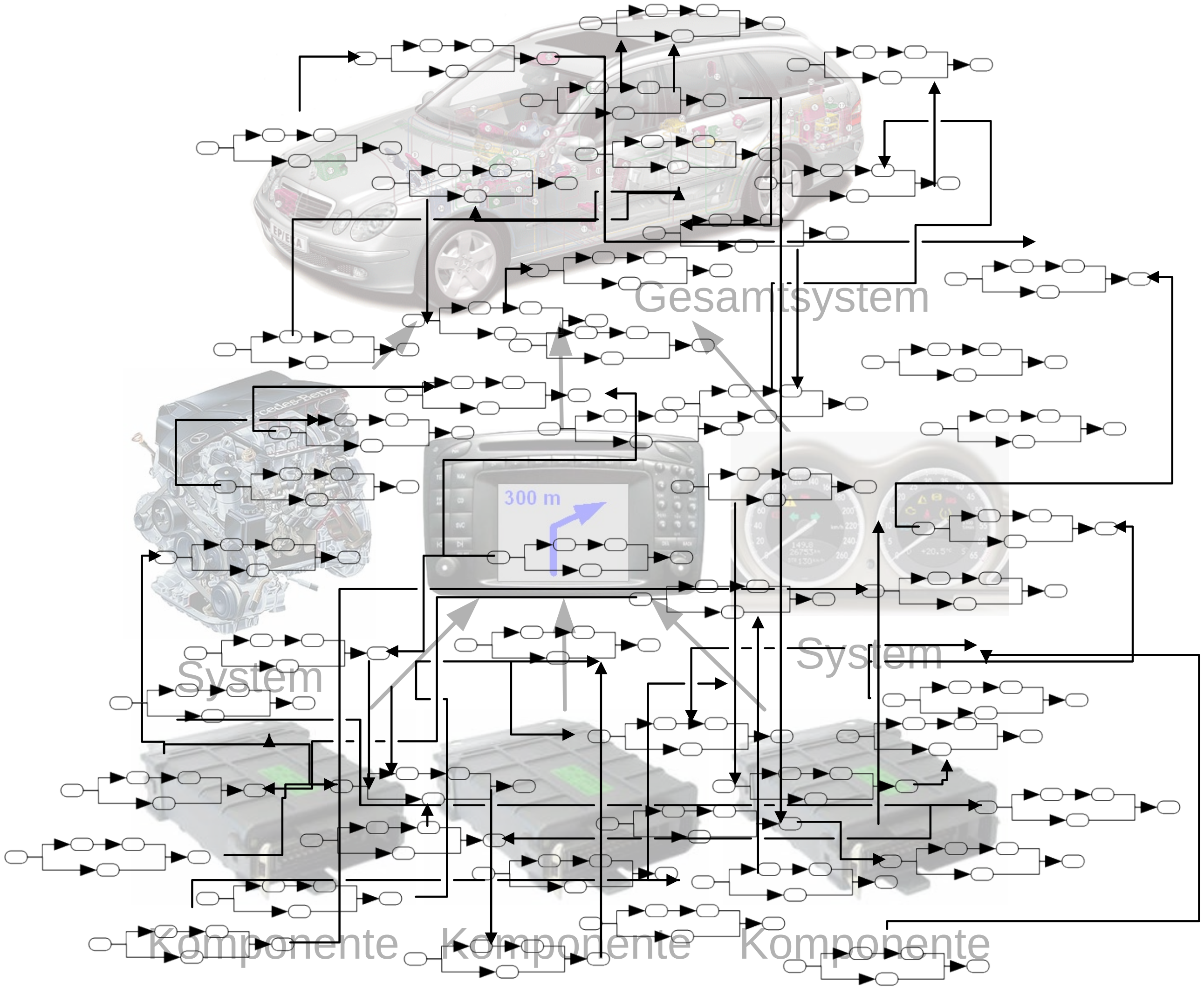




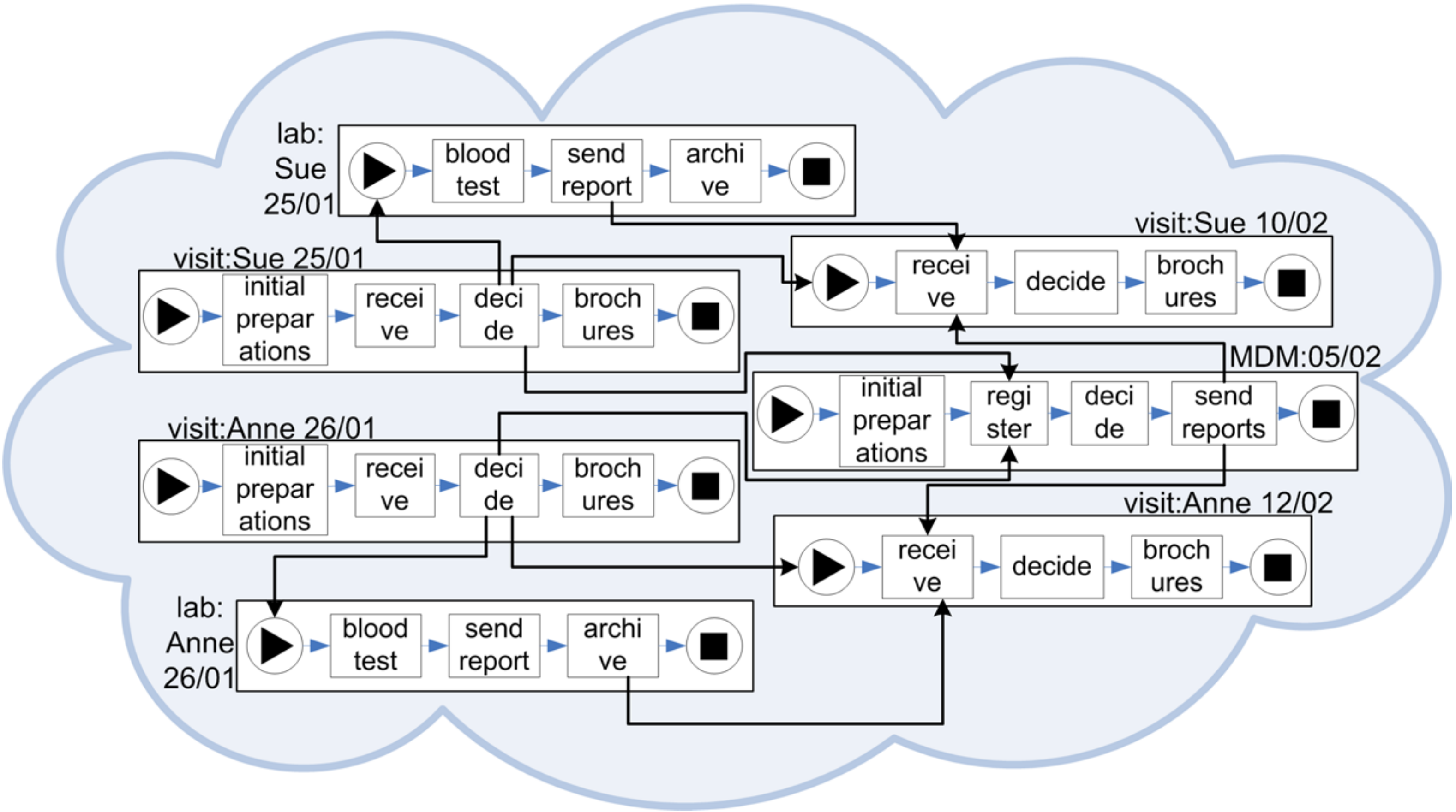
Agenda

- Challenges & Basic Notions
- Part I: Large Process Models
- Part II: Large Process Model Collections
- Part III: Large Process Structures
- References

Large Process Structures
Motivation



Large Process Structures
A Simple Example



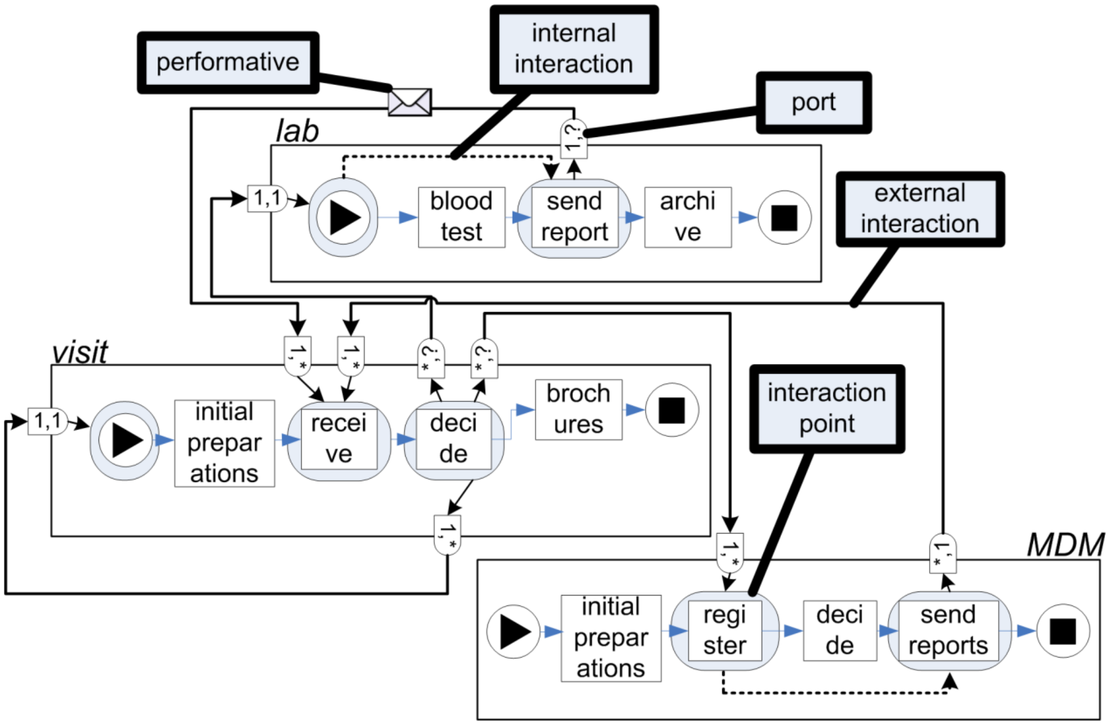
Interacting process fragments.
*The arcs show the **interactions** that need to take place between fragment instances.*

Mans et al., 2012

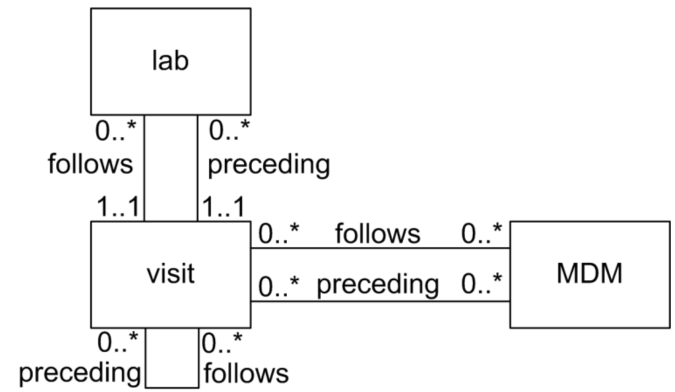
Large Process Structures
Proclets

- Proclets provide a framework for modeling and executing workflows
- A *Proclet* can be seen as a lightweight workflow process able to interact with other Proclets (potentially at different levels of aggregation).
- A *Proclet class* specifies which tasks need to be executed and in which order, i.e., the Proclet class defines the process followed by individual Proclets. One instance is called a *Proclet instance*.

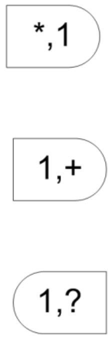
Large Process Structures
Proclets



a) visit, lab, and MDM Proclet classes



b) class diagram containing the three Proclet classes



output port with cardinality * (zero or more recipients) and multiplicity 1 (precisely one occurrence during the lifetime of the Proclet)

output port with cardinality 1 (precisely one recipient) and multiplicity + (at least one occurrence during the lifetime of the Proclet)

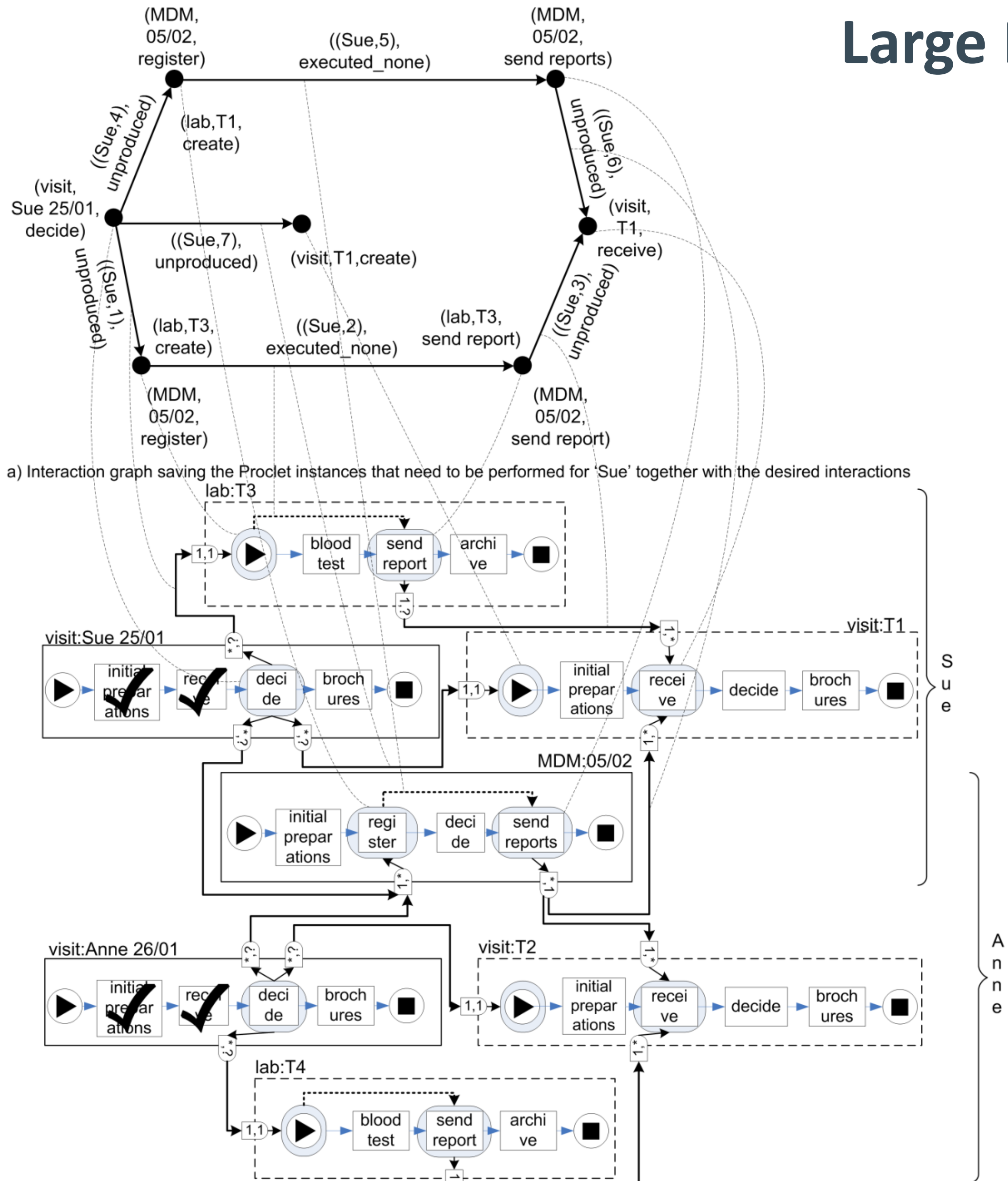
input port with cardinality 1 and multiplicity ? (at most one occurrence during the lifetime of the Proclet)

c) examples of port attributes

***Proclet classes** for the illustrated scenario*

Mans et al., 2012

Large Process Structures
Proclets

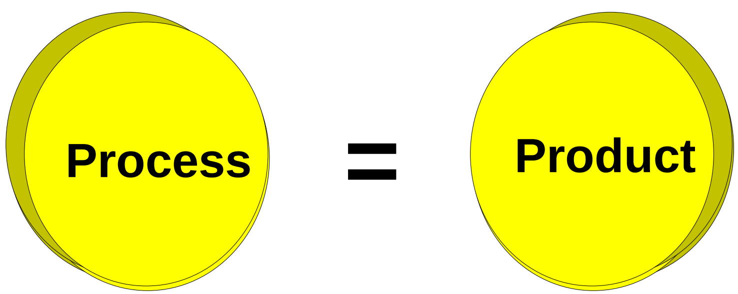


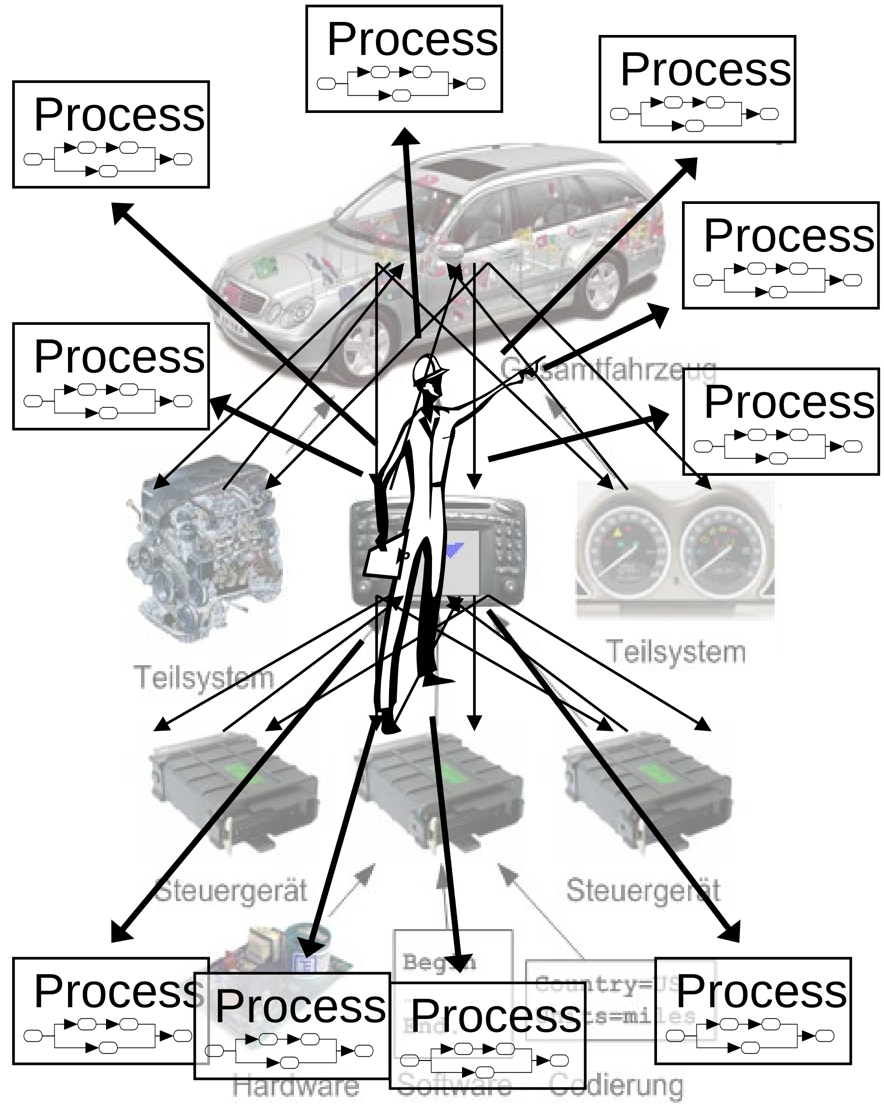
a) Interaction graph saving the Proclet instances that need to be performed for 'Sue' together with the desired interactions

b) The Proclet instances that need to be performed for 'Sue' and 'Anne'. For both the desired interactions are shown. Furthermore, for 'Sue' the instances that need to be performed are linked with the interaction graph via dotted arcs

Mans et al., 2012

Large Process Structures
Data-driven Process Structures: Motivation





Process structure needs to be adapted when product structure changes!

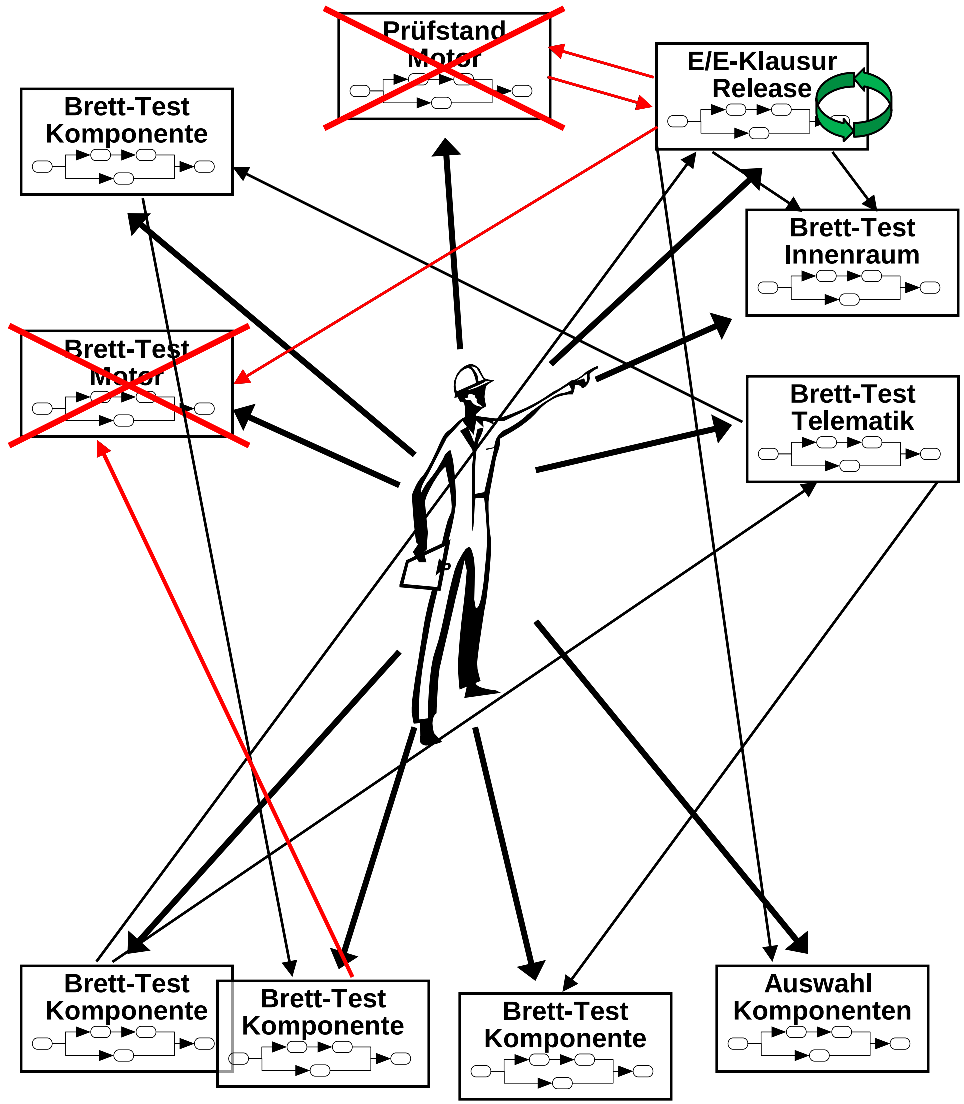
Large Process Structures
Data-driven Process Structures: Motivation

Modeling

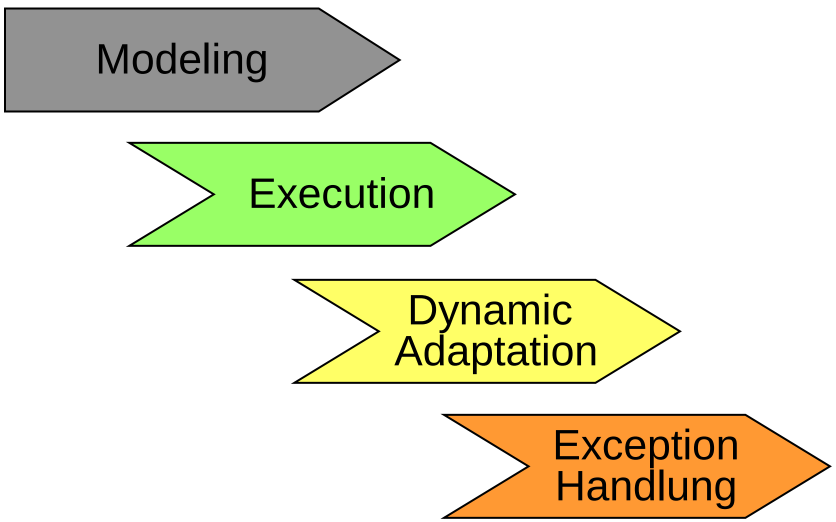
Execution

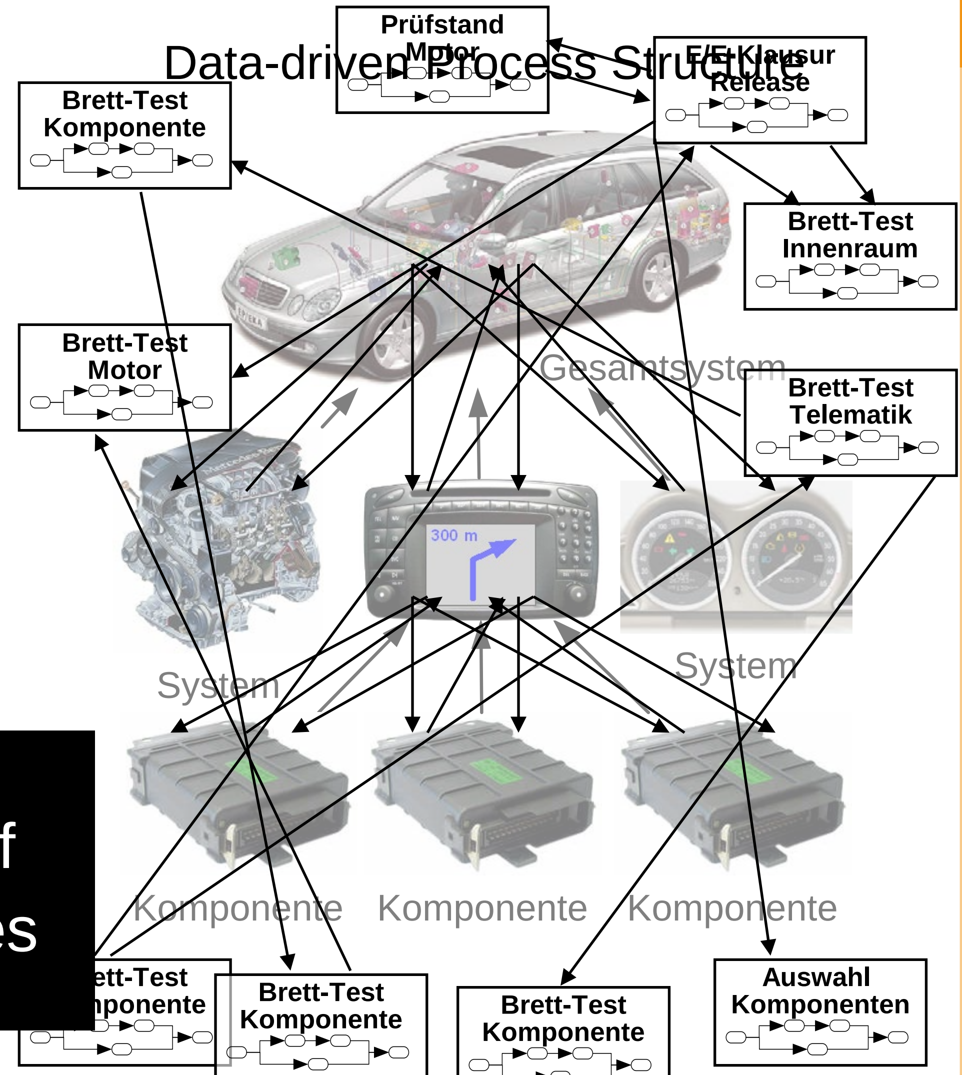
Dynamic Adaptation

Exception Handling



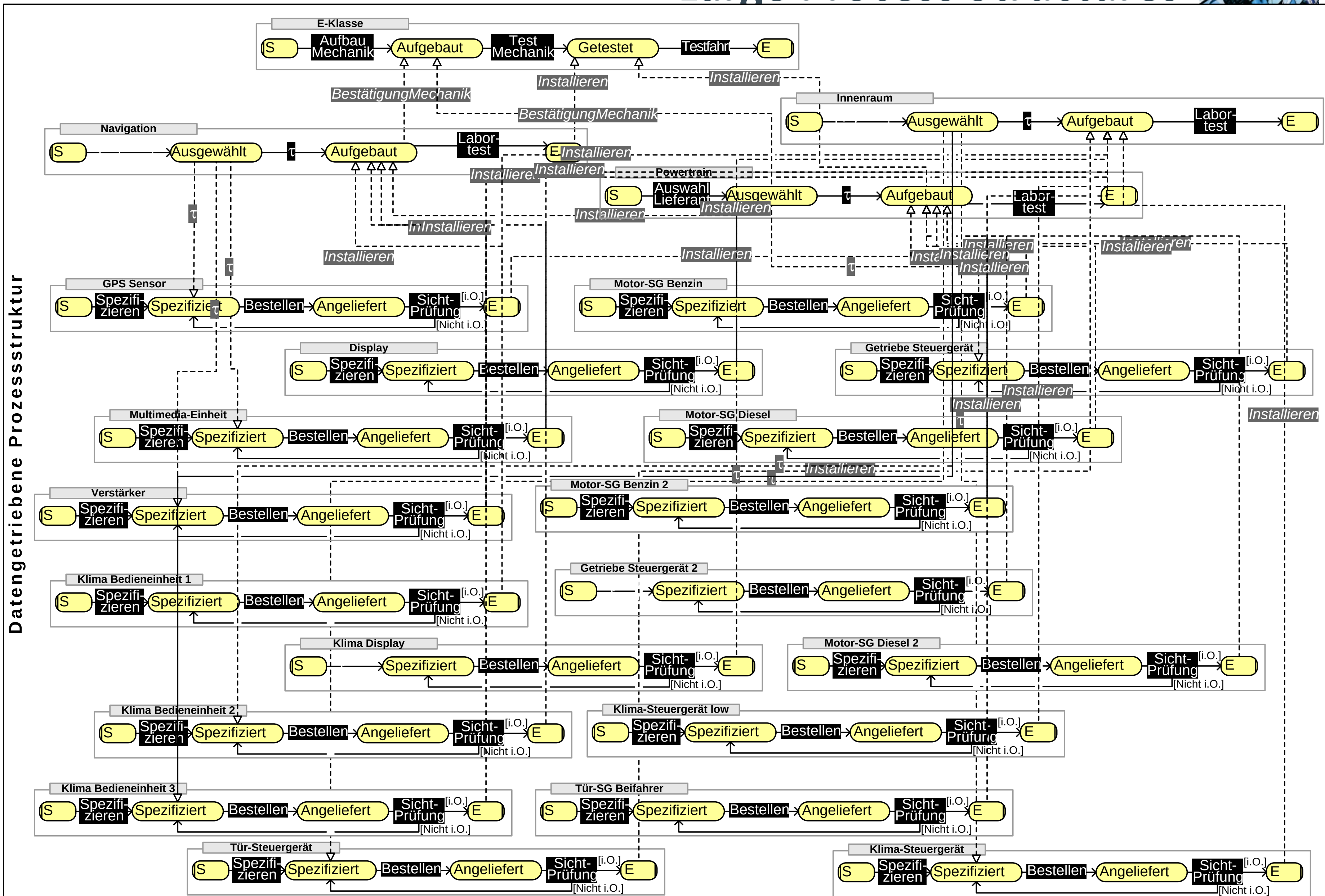
Large Process Structures
Data-driven Process Structures: Motivation





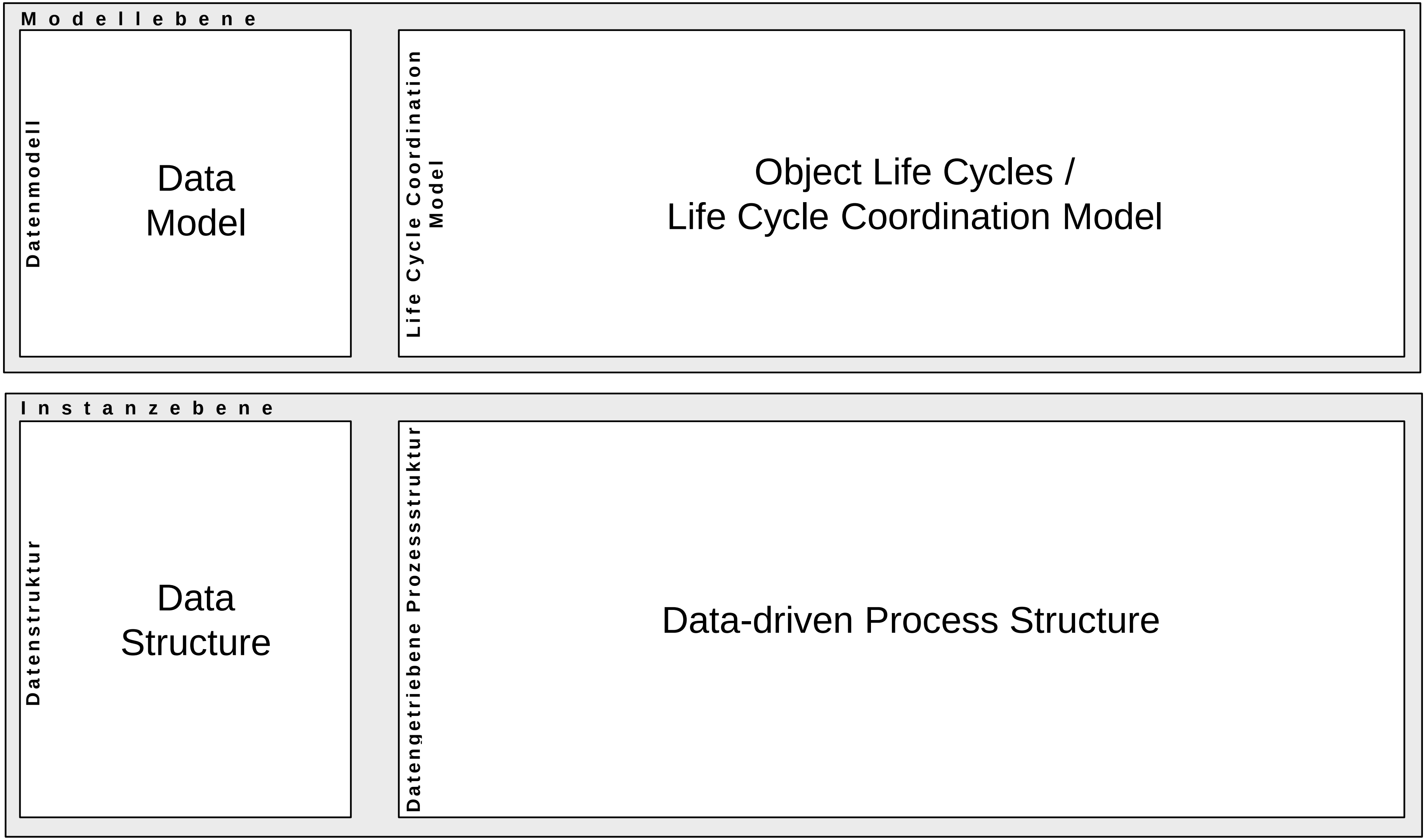
Corepro: Integrated Support of Data-driven Process Structures

Large Process Structures



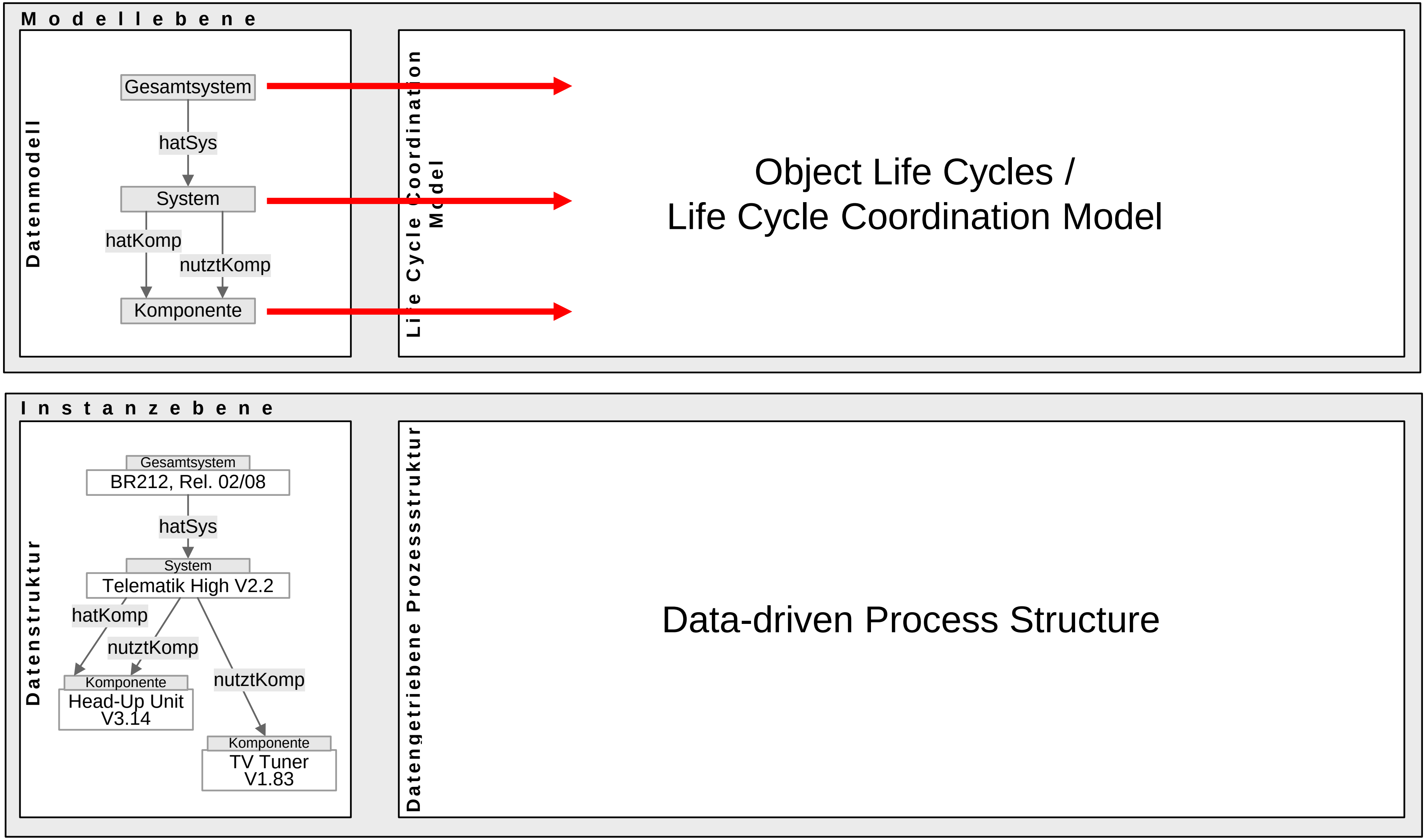
Data-Driven Process Structures:
The Corepro Approach

Large Process Structures
Corepro



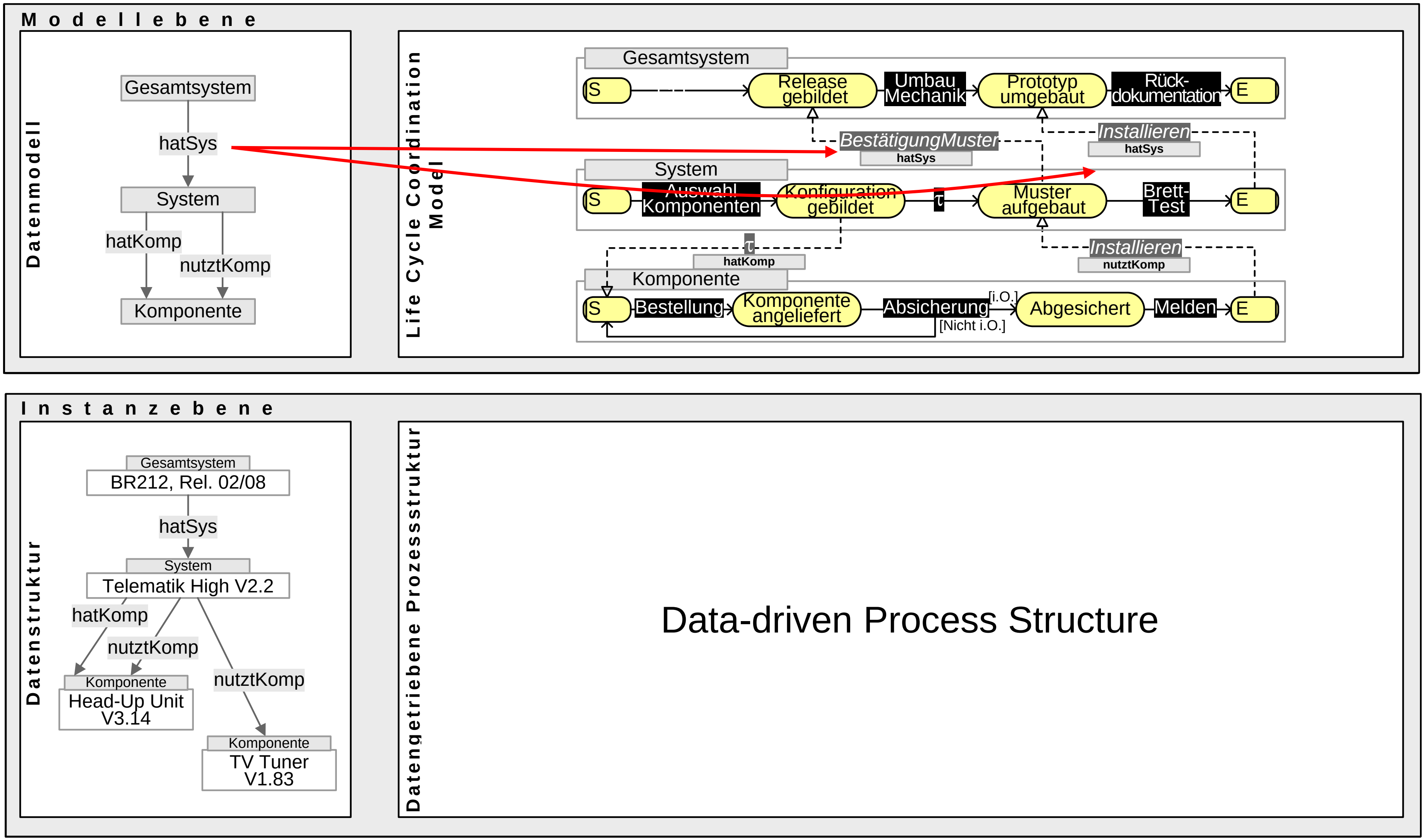
Müller et al. 2007

Large Process Structures
Corepro



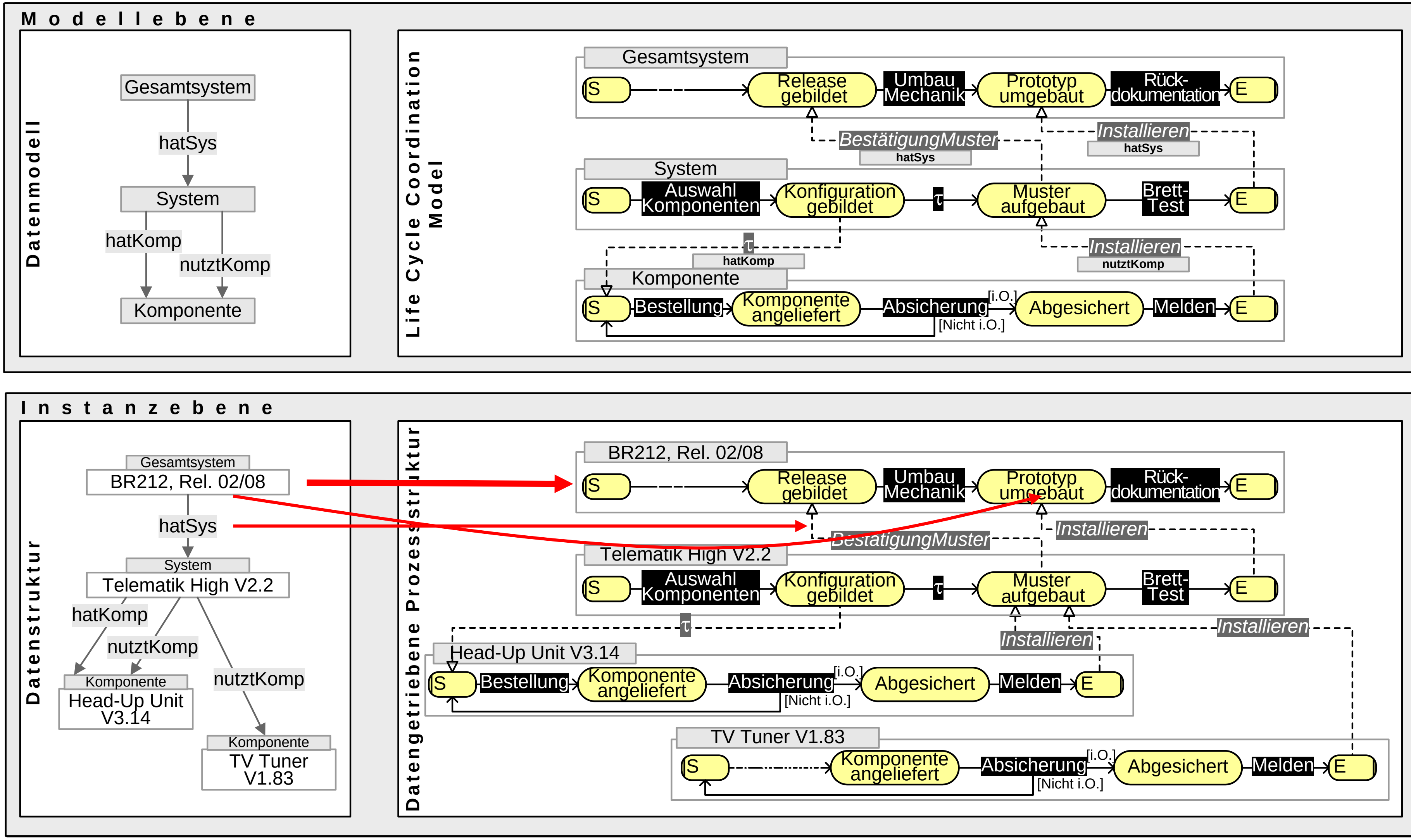
Müller et al. 2007

Large Process Structures
Corepro



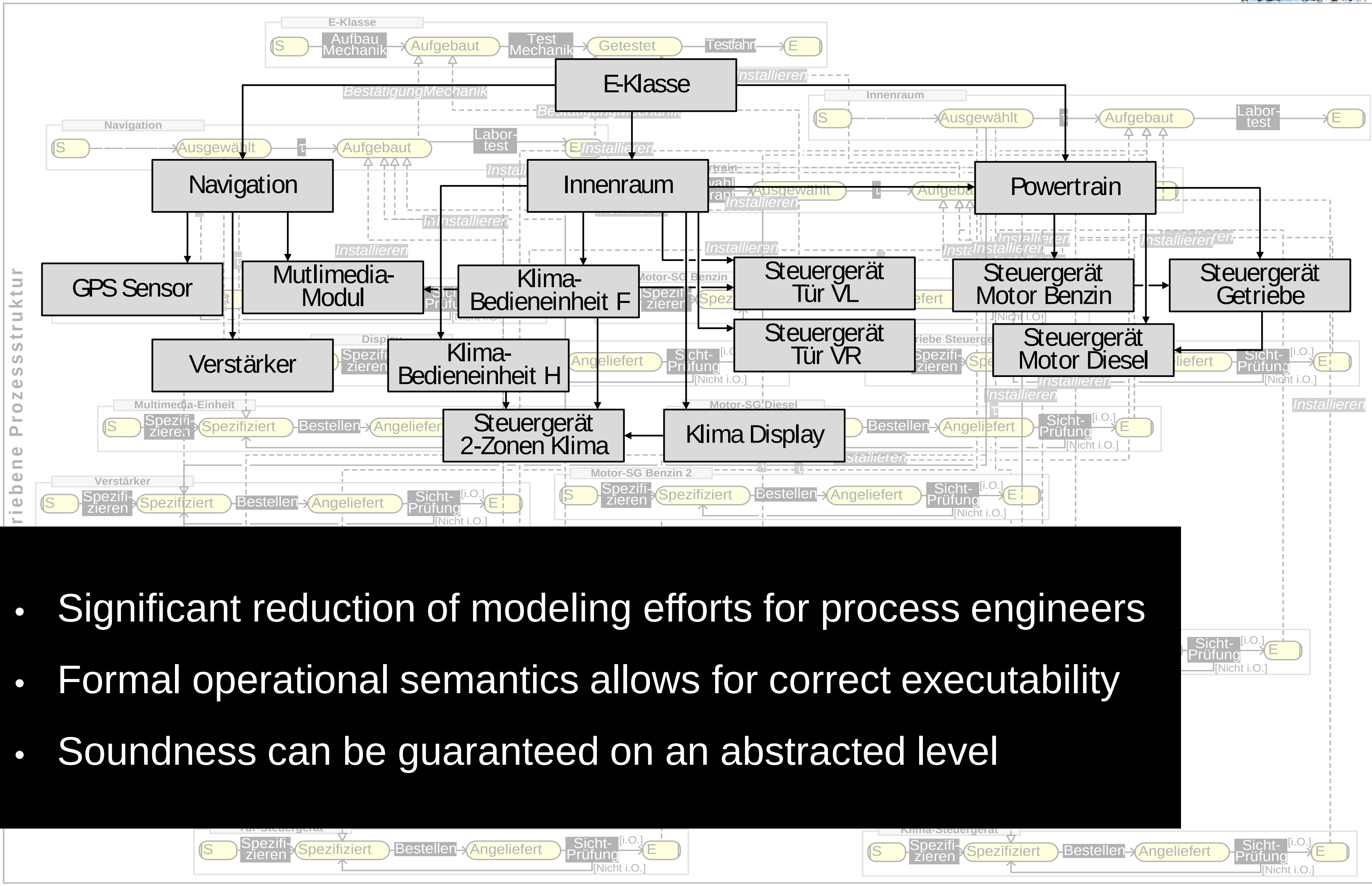
Müller et al. 2007

Large Process Structures
Corepro



Müller et al. 2007

Large Process Structures

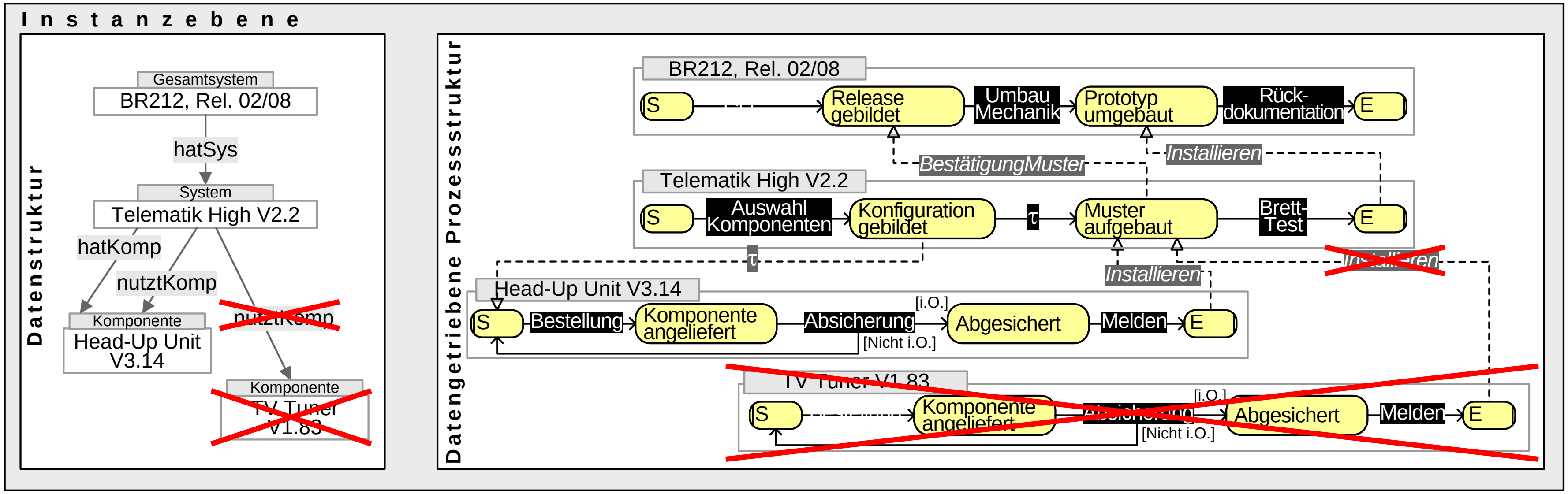


- Significant reduction of modeling efforts for process engineers
- Formal operational semantics allows for correct executability
- Soundness can be guaranteed on an abstracted level

Data-driven Process Adaptation

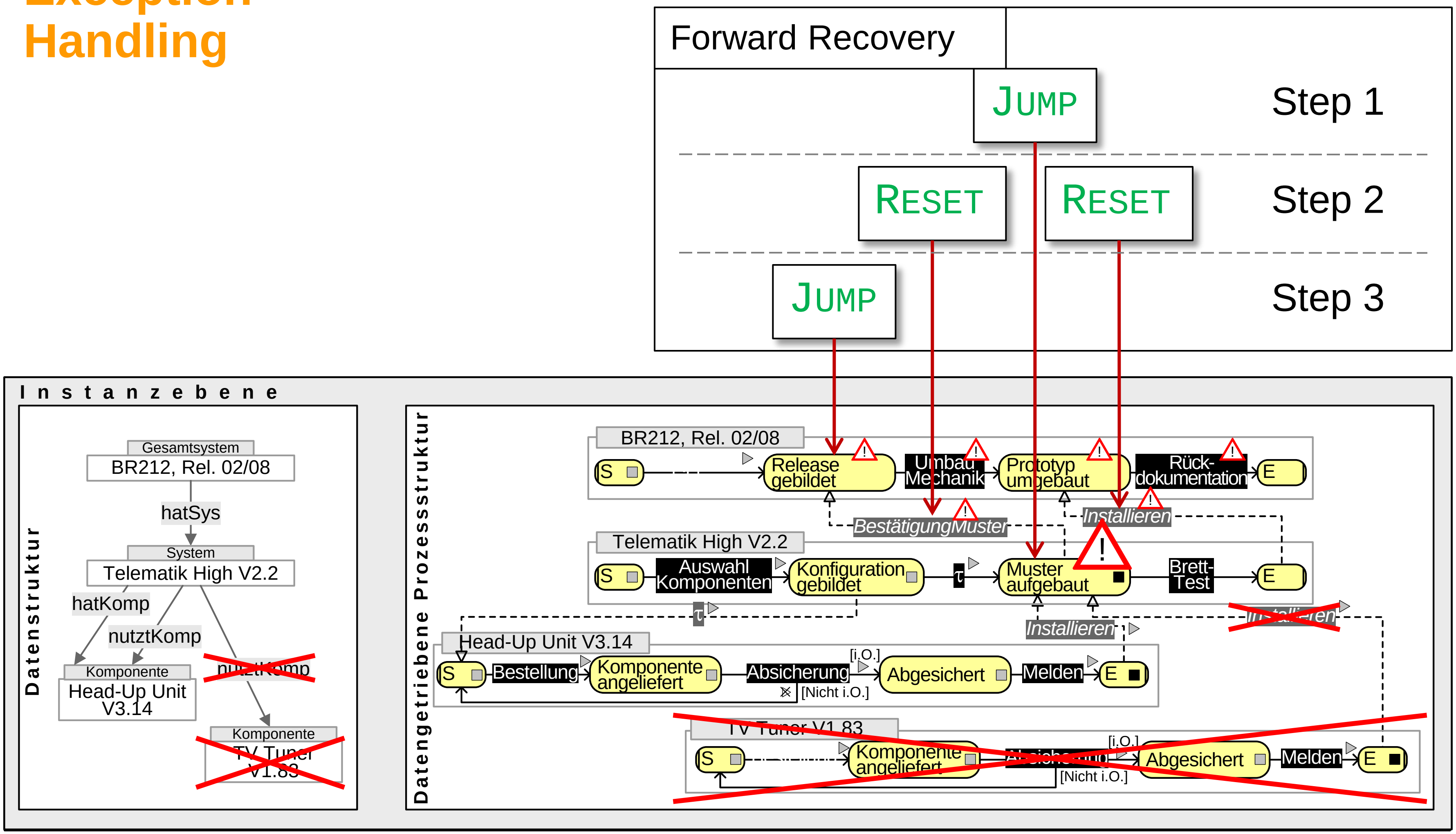
Müller et al. 2008a

Change Operation (Data Structure)	Change Operation (Process Structure)
1) removeRelation(Telematik High V2.2, TV Tuner V1.83, nutztKomp); 2) removeObject(TV Tuner V1.83);	1) removeExtTrans(Telematik High V2.2 . Muster Aufgebaut, Installieren, TV Tuner V1.83 . E); 2) removeOLC(Tuner V1.83);



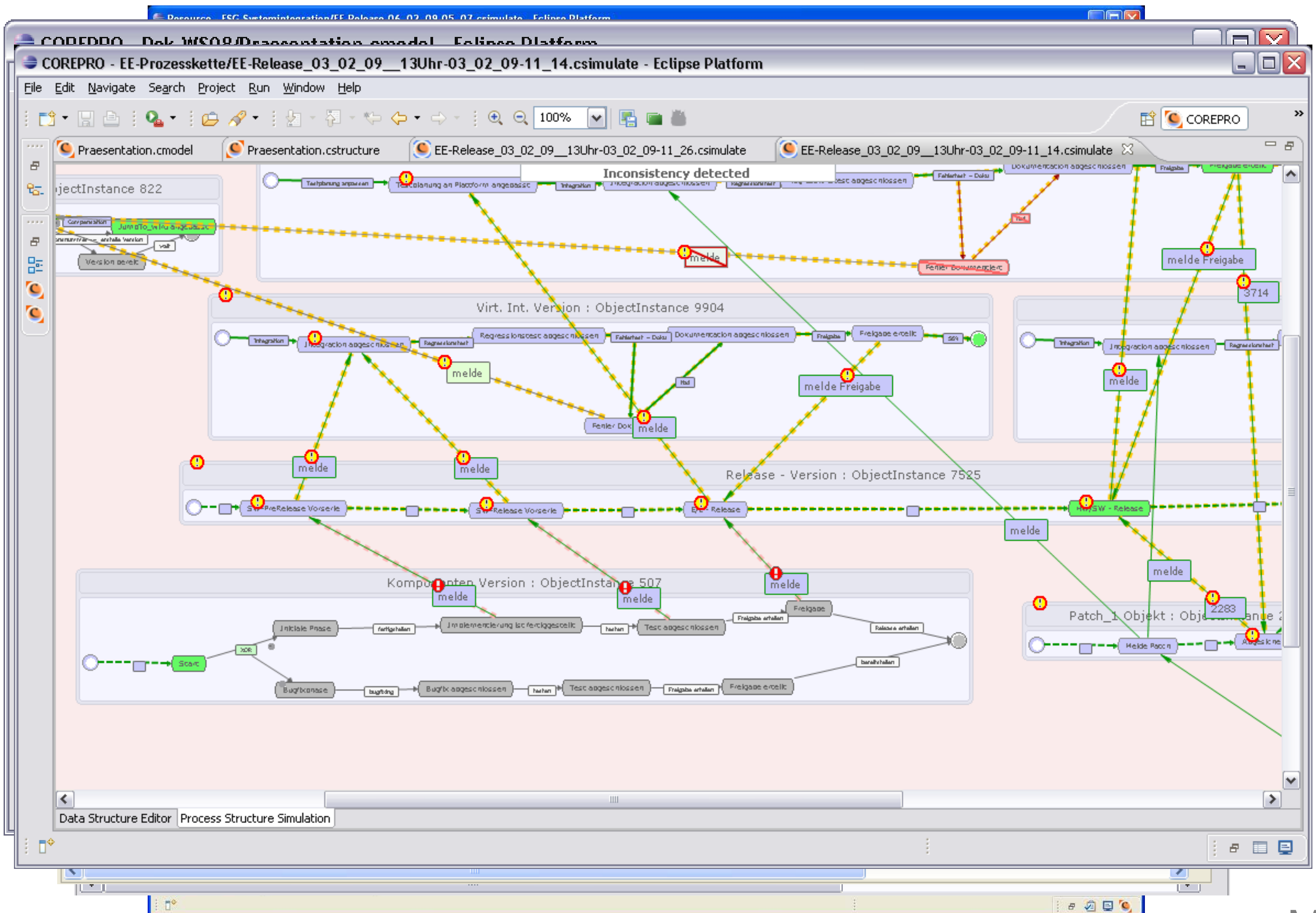
Data-driven Process Structures: Corepro

Exception Handling



Large Process Structures

Data-driven Process Structures: Corepro Proof-Of-Concept

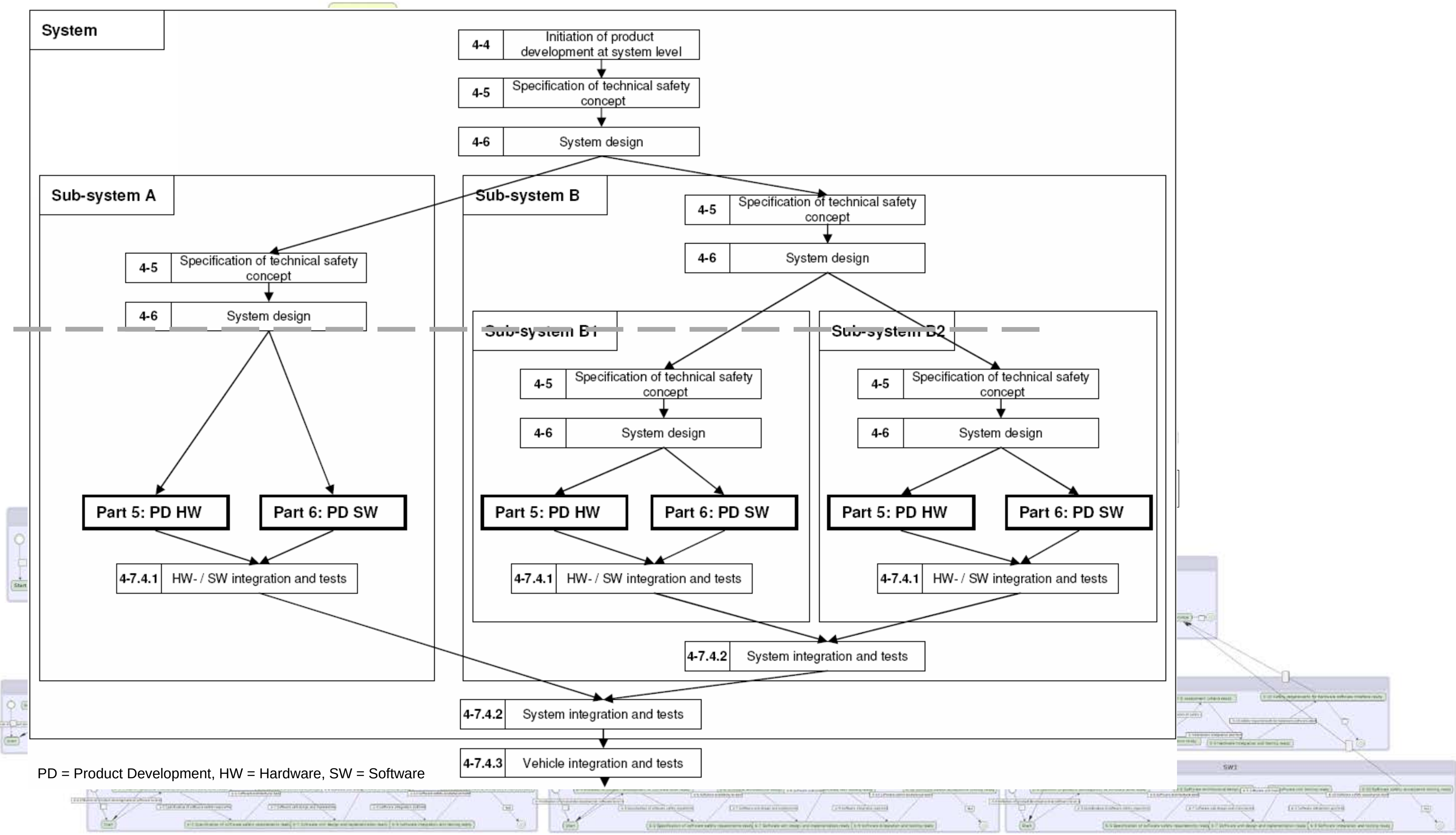


Müller et al. 2008b

Large Process Structures

Corepro: Case Study ISO 26262 -- Road Vehicles, Functional Safety

Instance Level: Da[illegible]ed Process Structure



PD = Product Development, HW = Hardware, SW = Software

References

Ahmed Awad: BPMN-Q: A Language to Query Business Processes. EMISA 2007: 115-128.

Ahmed Awad, Gero Decker, Mathias Weske: Efficient Compliance Checking Using BPMN-Q and Temporal Logic. BPM 2008: 326-341.

Bobrik, R.; Reichert, M.; Bauer, T.: Requirements for the Visualization of System-Spanning Business Processes. Proc 16th Int'l Workshop on Database and Expert Systems Applications (DEXA'05), IEEE Computer Society, pp. 948–954 (2005)

Bobrik, R.; Bauer, T.; Reichert, M.: Proviado – Personalized and Configurable Visualizations of Business Processes. Proc 7th Int'l Conf on Electronic Commerce and Web Technologies (EC-Web'06) LNCS 4082, pp. 61–71 (2006)

Bobrik, R.; Reichert, M.; Bauer, Th.: View-based Process Visualization. In: Proc Int'l Conf on Business Process Management (BPM'07) LNCS 4714, pp. 88–95 (2007)

Remco M. Dijkman, Marlon Dumas, Boudewijn F. van Dongen, Reina Käärik, Jan Mendling: Similarity of business process models: Metrics and evaluation. Inf. Syst. 36(2): 498-516 (2011)

M. Dumas, L. Garcia-Banuelos, M. La Rosa, R. Uba. Fast Detection of Exact Clones in Repositories of Business Process Models. *Information Systems* (to appear).

Eshuis, R., Grefen, P.: Constructing Customized Process views. Data Knowl Eng, 64, pp. 419-438 (2008)

Christian Gerth, Markus Luckey, Jochen Malte Küster, Gregor Engels: Detection of Semantically Equivalent Fragments for Business Process Model Change Management. IEEE SCC 2010: 57-64

D. Fahland: Number of ways students name an activity in a process model. http://dirksmetric.wordpress.com/2012/08/17/number-of-ways-student-name-an-activity-in-a-process-model/, [accessed on 2012/08/12]

Figl, K., Laue, R.: Cognitive Complexity in Business Process Modeling. In Proc. CAiSE'11, 452-466.

Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H., La Rosa, M.: Configurable Workflow Models. International Journal of Cooperative Information Systems 17(2): 177-221 (2008).

Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach, J. Soft. Maintenance. 22(6-7): 519-546 (2010)

Hipp, Markus and Mutschler, Bela and Reichert, Manfred (2012) *Navigating in Complex Business Processes.* In: Proc. 23rd Int'l Conf on Database and Expert Systems Applications (DEXA'12), Part II, Vienna, Austria, September 3-6, 2012, LNCS 7447, Springer, pp. 466-480.

Kolb, J.; Kammerer, K.; Reichert, M. : Updatable Process Views for User-centered Adaption of Large Process Models. Proc 10th Int'l Conf on Service Oriented Computing (ICSOC'12), (2012) Kumar, A., Wen, Y.: Design and management of flexible process variants using templates and rules. International Journal Computers in Industry 63(2), pp. 112-130 (2012).

La Rosa, M., Dumas, M., ter Hofstede, A.H.M. , Mendling, J., Gottschalk, F.: Beyond Control-Flow: Extending Business Process Configuration to Roles and Objects. In Proc. ER'08, 199-215.

La Rosa, M., Dumas, M., Uba, R., Dijkman, R.M.: Merging business process models. In: OTM Confs (1), LNCS, vol. 6426, pp. 96–113. Springer (2010)

Marcello La Rosa, Hajo A. Reijers, Wil M. P. van der Aalst, Remco M. Dijkman, Jan Mendling, Marlon Dumas, Luciano García-Bañuelos: APROMORE: An advanced process model repository. Expert Syst. Appl. 38(6): 7029-7040 (2011)

Larkin, J.H., Simon, H.A.: Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11 (1987) 65-100

Henrik Leopold, Jan Mendling, Hajo A. Reijers: On the Automatic Labeling of Process Models. CAiSE 2011: 512-520

Henrik Leopold, Sergey Smirnov, Jan Mendling: Recognising Activity Labeling Styles in Business Process Models. Enterprise Modelling and Information Systems Architectures 6(1): 16-29 (2011)

Henrik Leopold, Sergey Smirnov, Jan Mendling: On the refactoring of activity labels in business process models. Inf. Syst. 37(5): 443-459 (2012)

Müller, D.; Reichert, M.; Herbst, J.: Data-driven Modeling and Coordination of Large Process Structures. In: 15th Int'l Conf. on Cooperative Information Systems (CoopIS'07), Vilamoura, Portugal, LNCS 4803, Springer, pp. 131-149 (2007).

Müller, D.; Reichert, M.; Herbst, J.: A New Paradigm for the Enactment and Dynamic Adaptation of Data-driven Process Structures. In: 20th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'08), Montpellier, France, LNCS 5074, Springer, pp. 48-63 (2008a)

Müller, D.; Reichert, M.; Herbst, J.; Köntges, D.; Neubert, A.: COREPRO-Sim: A Tool for Modeling, Simulating and Adapting Data-driven Process Structures. In: 6th Int'l Conf. on Business Process Management (BPM'08 Demonstrations), Milan, Italy, LNCS 5240, Springer, pp. 394-397 (2008b)

Li, C., Reichert, M., Wombacher, A. (2011) *Mining Business Process Variants: Challenges, Scenarios, Algorithms.* Data & Knowledge Engineering, Vol. 70, No. 5, pp. 409-434.

T. Malone, K. Crowston, G. Herman, Organizing business knowledge: the MIT process handbook, MIT Press, 2003.

Mans, R.; van der Aalst, W.; Russel, N.; Bakker, P.; Moleman, A.: Lightweight Interacting Patient Treatment Processes. International Journal of Knowledge-Based Organizations (to appear in 2012)

Jan Mendling, Hajo A. Reijers, Jan Recker: Activity labeling in process modeling: Empirical insights and recommendations. Inf. Syst. 35(4): 467-482 (2010)

Jan Mendling, Hajo A. Reijers, Wil M. P. van der Aalst: Seven process modeling guidelines (7PMG). Information & Software Technology 52(2): 127-136 (2010)

J. Mendling, H. Verbeek, B.F. van Dongen, W.M.P. van der Aalst, G. Neumann, Detection and prediction of errors in EPCs of the SAP reference model, Data & Knowledge Engineering 64 (1) (2008) 312–329.

Miller, G.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review 63 (1956) 81-97.

Paas, F., Tuovinen, J.E., Tabbers, H., Gerven, P.W.M.V.: Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. Educational Psychologist 38 (2003) 63-71.

Parnas, D.L.: On the Criteria to be Used in Decomposing Systems into Modules. Communications of the ACM 15 (1972) 1053-1058.

Artem Polyvyanyy, Luciano García-Bañuelos, Marlon Dumas: Structuring acyclic process models. Inf. Syst. 37(6): 518-538 (2012)

Reichert, M.; Kolb, J.; Bobrik, R.; Bauer, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. Proc 27th ACM Symposium on Applied Computing (SAC'12), ACM Press, pp. 1653-1660 (2012)

Reichert, M.: Visualizing Large Business Process Models: Challenges, Techniques, Applications. Proc. BPM'12 Workshops, 1st Int'l Workshop on Theory and Applications of Process Visualization (TAProViz'12), LNBIP (2012) – to appear!

Hajo A. Reijers, Jan Mendling, Remco M. Dijkman: Human and automatic modularizations of process models to enhance their comprehension. Inf. Syst. 36(5): 881-897 (2011)

Rosemann, M., van der Aalst, W.M.P.: A configurable reference modeling language. Information Systems 32(1), pp. 1-23 (2007).

Schumm, D.; Latuske, G.; Leymann, F. et al.: State Propagation for Business Process Monitoring on Different Levels of Abstraction. Proc 19th ECIS (2011)

Sharp A., McDermott P. (2008) Workflow Modeling: Tools for Process Improvement and Applications Development. Artech House Publishers, Norwood

Smirnov, S.; Reijers, H.; Weske, M.: A Semantic Approach for Business Process Model Abstraction. In: Advanced Information Systems Engineering, pp. 497-511 (2011)

M. Soto, A. Ocampo, J. Munch, The secret life of a process description: a look into the evolution of a large process model, in: Proc. ICSP'08, 2008.

Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive Science 12 (1988) 257-285.

Sweller, J., Chandler, P.: Why Some Material Is Difficult to Learn. Cognition and Instruction 12 (1994) 185-233.

Barbara Weber, Manfred Reichert: Refactoring Process Models in Large Process Repositories. CAiSE 2008: 124-139

Barbara Weber, Manfred Reichert, Jan Mendling, Hajo A. Reijers: Refactoring large process model repositories. Computers in Industry 62(5): 467-486 (2011)

Z. Yan, R. Dijkman, P. Grefen: FNet: An Index for Advanced Business Process Querying. Proc. BPM’13, pp. 246-261.

S. Zugal, J. Pinggera, J. Mendling, H. Reijers and B. Weber: Assessing the Impact of Hierarchy on Model Understandability-A Cognitive Perspective. In: Proc. EESSMod ’11, pp. 123–133, 2011.

S. Zugal, J. Pinggera and B. Weber: The Impact of Testcases on the Maintainability of Declarative Process Models. In: Proc. BPMDS ’11, pp. 163–177, 2011.

S. Zugal, J. Pinggera and B. Weber: Assessing Process Models with Cognitive Psychology. In: Proc. EMISA ’11, pp. 177–182, 2011.