Process model management and analytics

presented by
Barbara Weber

DTU Compute
Department of Applied Mathematics and Computer Science

Business Process

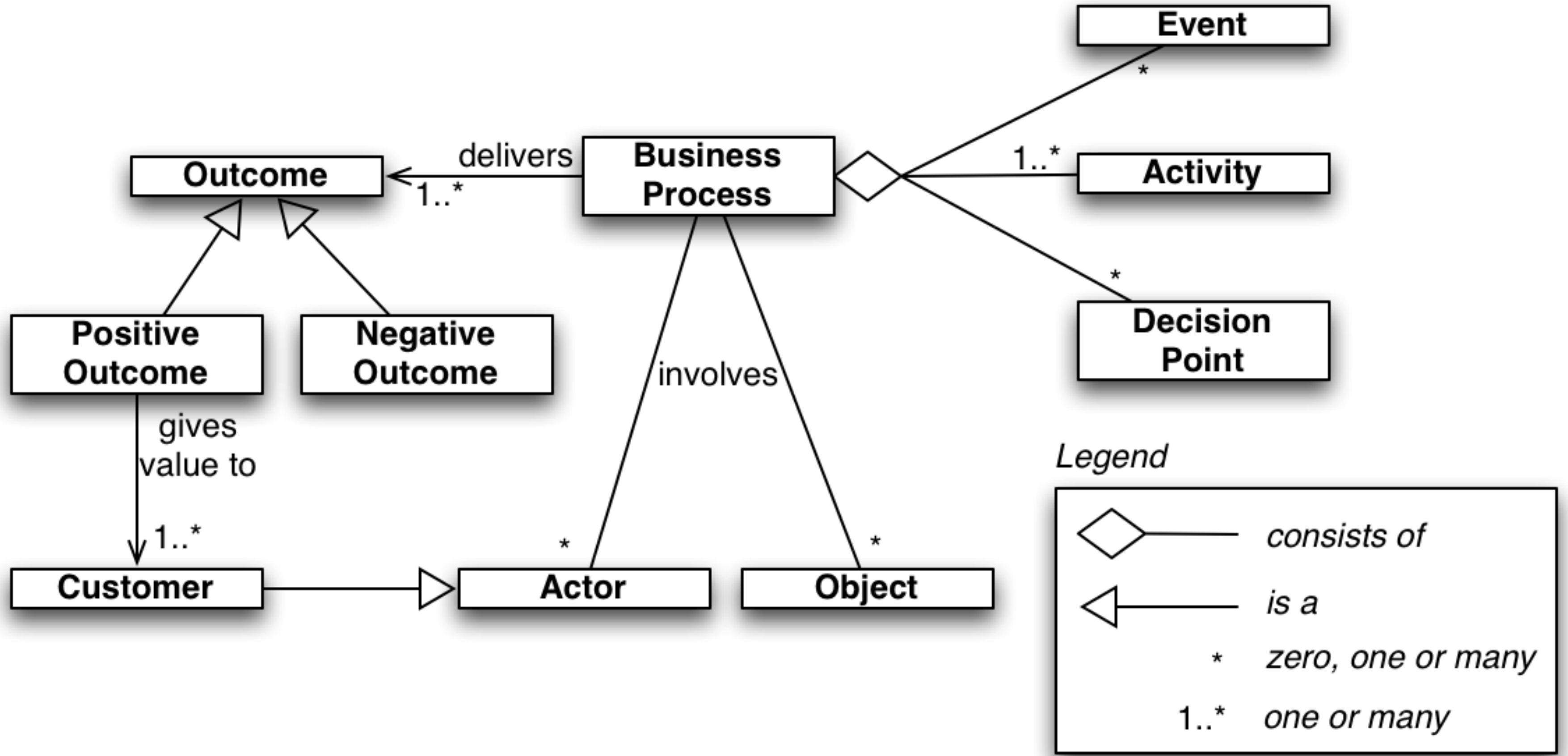
Collection of related events, activities and decisions, that involve a number of actors and objects, and that collectively lead to an outcome that is of value to an organization or its customers.

Examples:

- Order-to-Cash
- Quote-to-Order
- Procure-to-Pay
- Fault-to-Resolution (Issue-to-Resolution) /Claim-to-Settlement
- Application-to-Approval

Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers: Fundamentals of Business Process Management. Springer 2013, ISBN 978-3-642-33142-8, pp. I-XXVII, 1-399

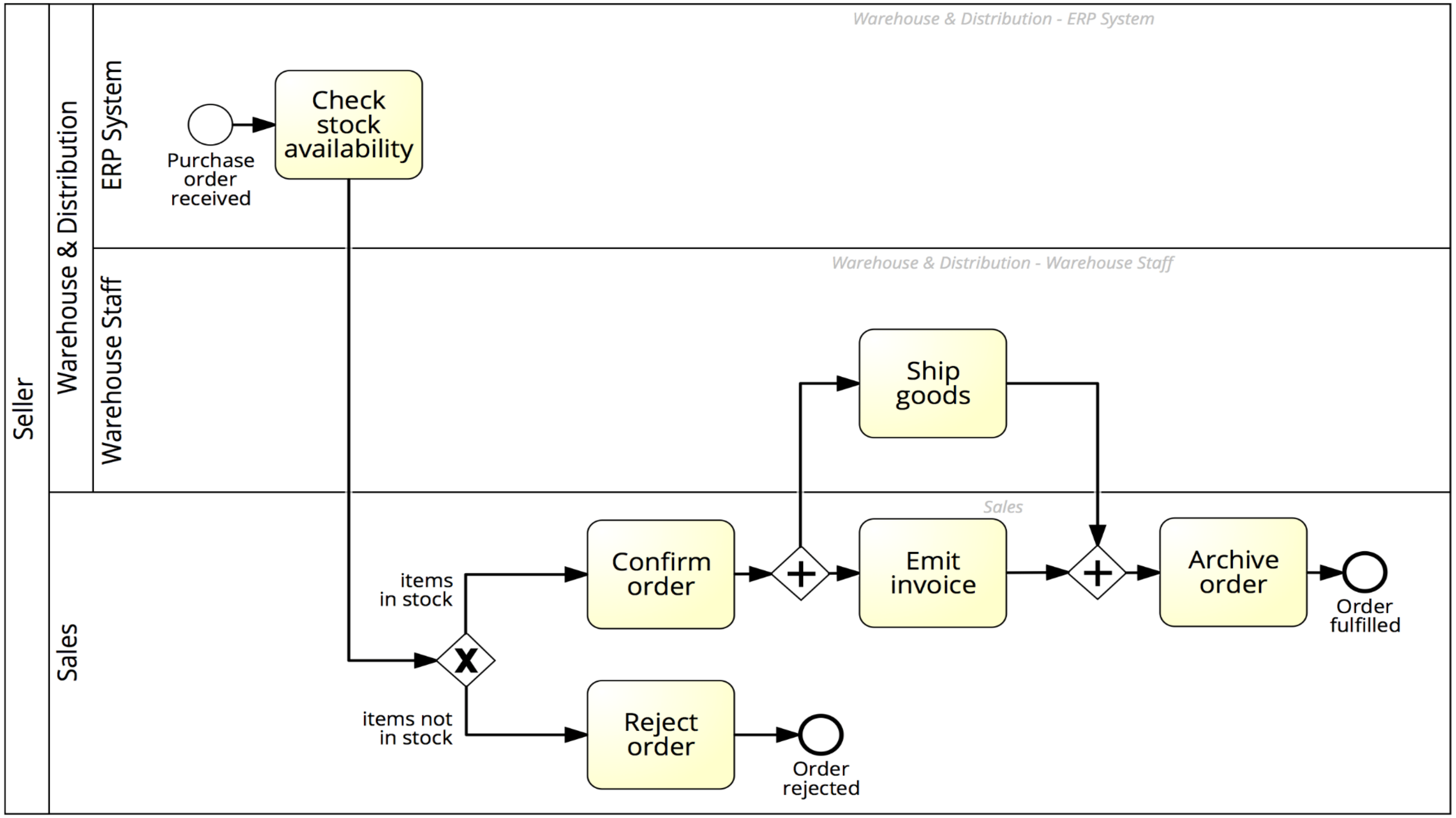
What is a Business Process: Recap



Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers: Fundamentals of Business Process Management. Springer 2013, ISBN 978-3-642-33142-8, pp. I-XXVII, 1-399

Process Model

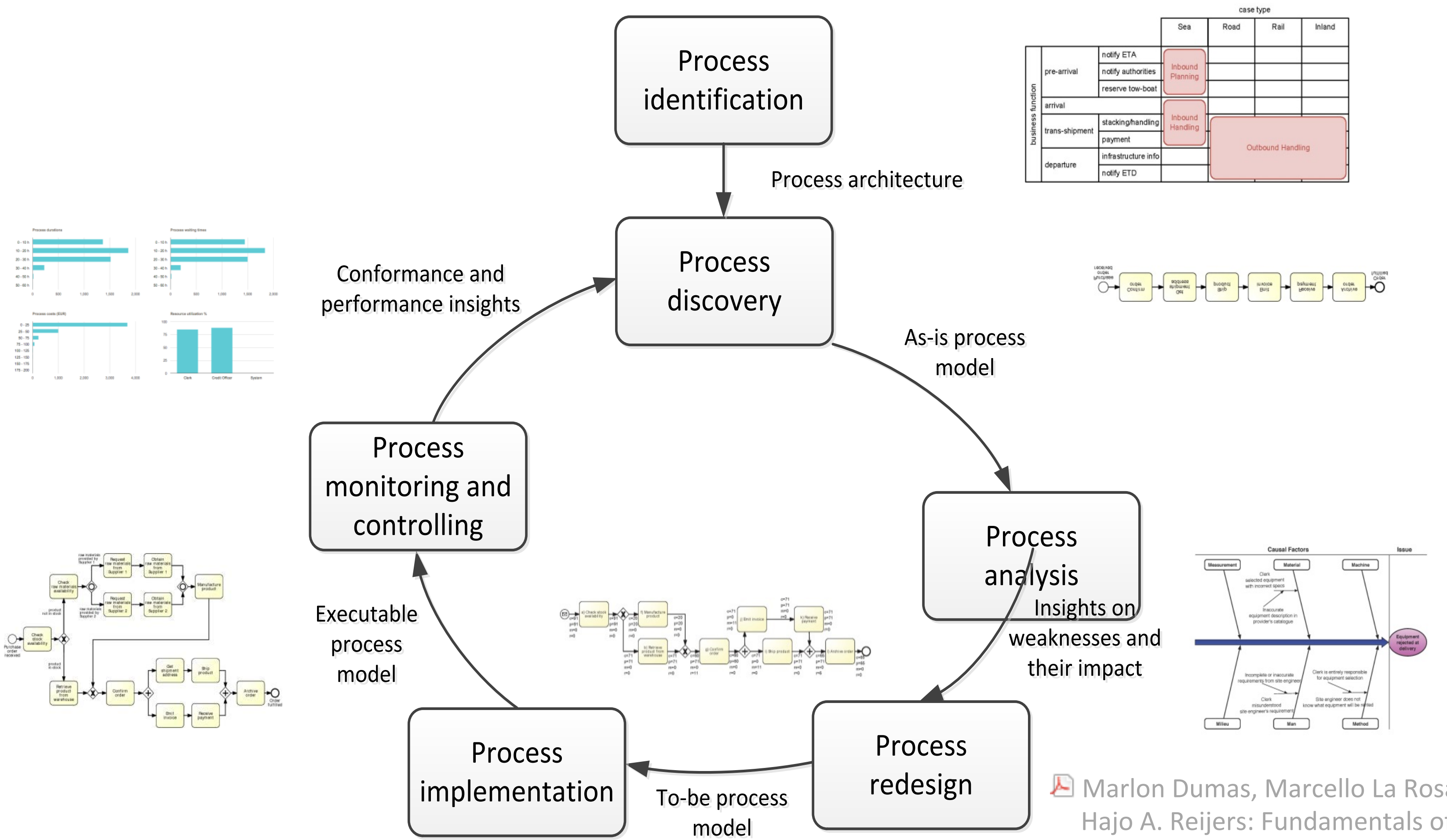
- Graphical representation of a business process



Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers: Fundamentals of Business Process Management. Springer 2013, ISBN 978-3-642-33142-8, pp. I-XXVII, 1-399

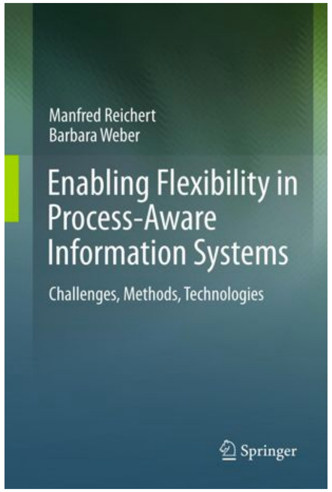
Business Process Management

…designing, analyzing, redesigning, executing, and monitor business processes.



Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers: Fundamentals of Business Process Management. Springer 2013, ISBN 978-3-642-33142-8, pp. I-XXVII, 1-399

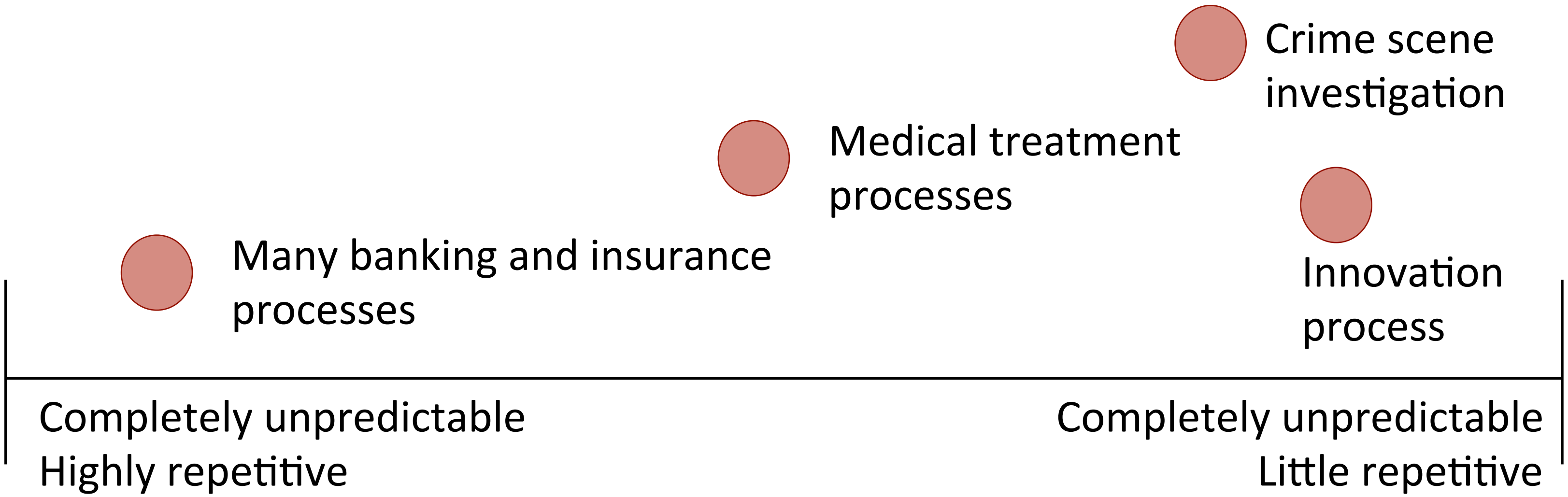
Enabling Flexibility
Process Adaptations, Process Evolution, and Variability



M. Reichert and B. Weber: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies, Springer 2012

The Process Spectrum

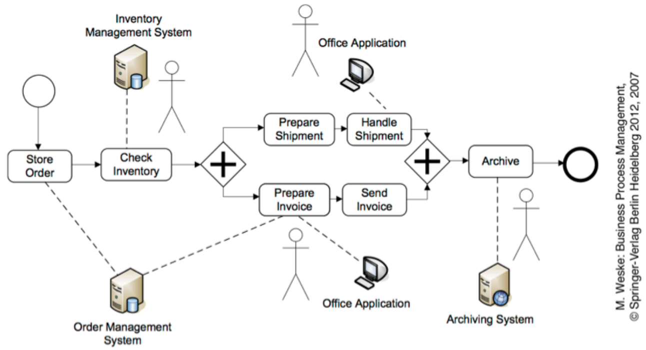
- The process spectrum reaches from
  - completely predictable and highly repetitive
  - to completely unpredictable and little repetitive



The Process Spectrum

- The process spectrum reaches from
  - completely predictable and highly repetitive
  - to completely unpredictable and little repetitive

Pre-specified process model, e.g., using BPMN



Completely unpredictable
Highly repetitive

Completely unpredictable
Little repetitive

The Process Spectrum

- The process spectrum reaches from
  - completely predictable and highly repetitive
  - to completely unpredictable and little repetitive

Process Adaptation

Process Evolution

Process Variability

Completely unpredictable
Highly repetitive

Completely unpredictable
Little repetitive

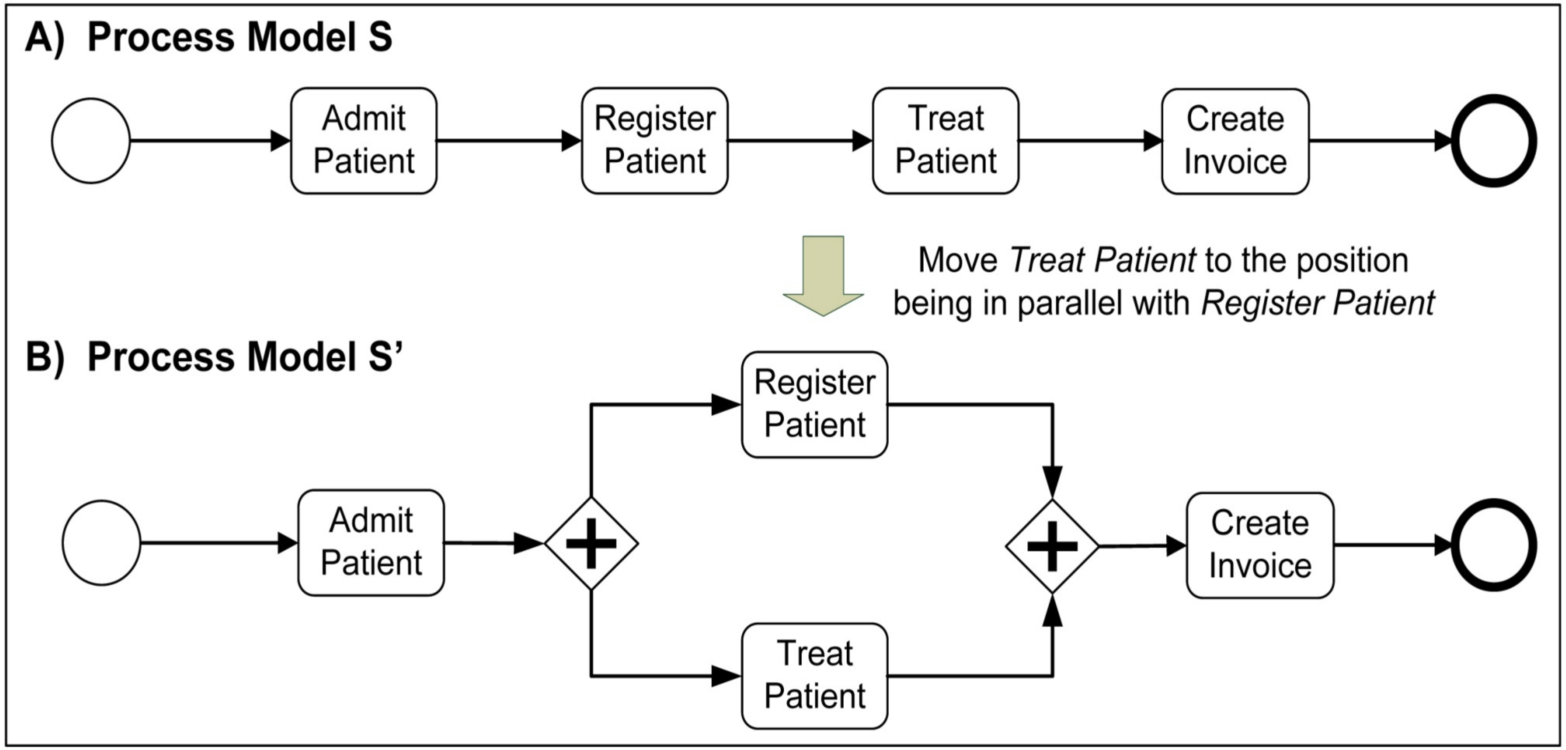
Process Adaption

- **Ability to adapt process and its structure to temporary events**
  (due to special cases, exceptions)

- Planned
  - Typically handled via exception handling
- Unplanned
  - Require ad-hoc changes, i.e., structural process model adaptations changes

Process Adaption through Ad-hoc Changes

- Behavioral changes require <u>structural process model adaptations</u>

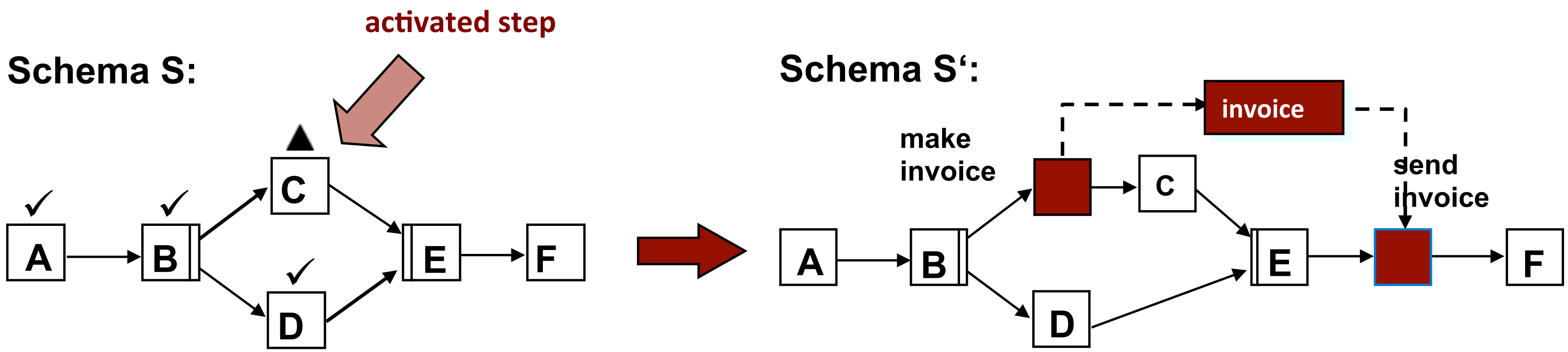


M. Reichert, B.Weber: Enabling Flexibility in Process-Aware Information Systems, @ Springer-Verlag Heidelberg 2012

- as well as <u>adaptations</u> of the <u>process instance state</u>

State Compliance
A Correctness Notion for Dynamic Instance Changes

Ensuring Dynamic Correctness



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

Need for general correctness criterion

⇨State Compliance

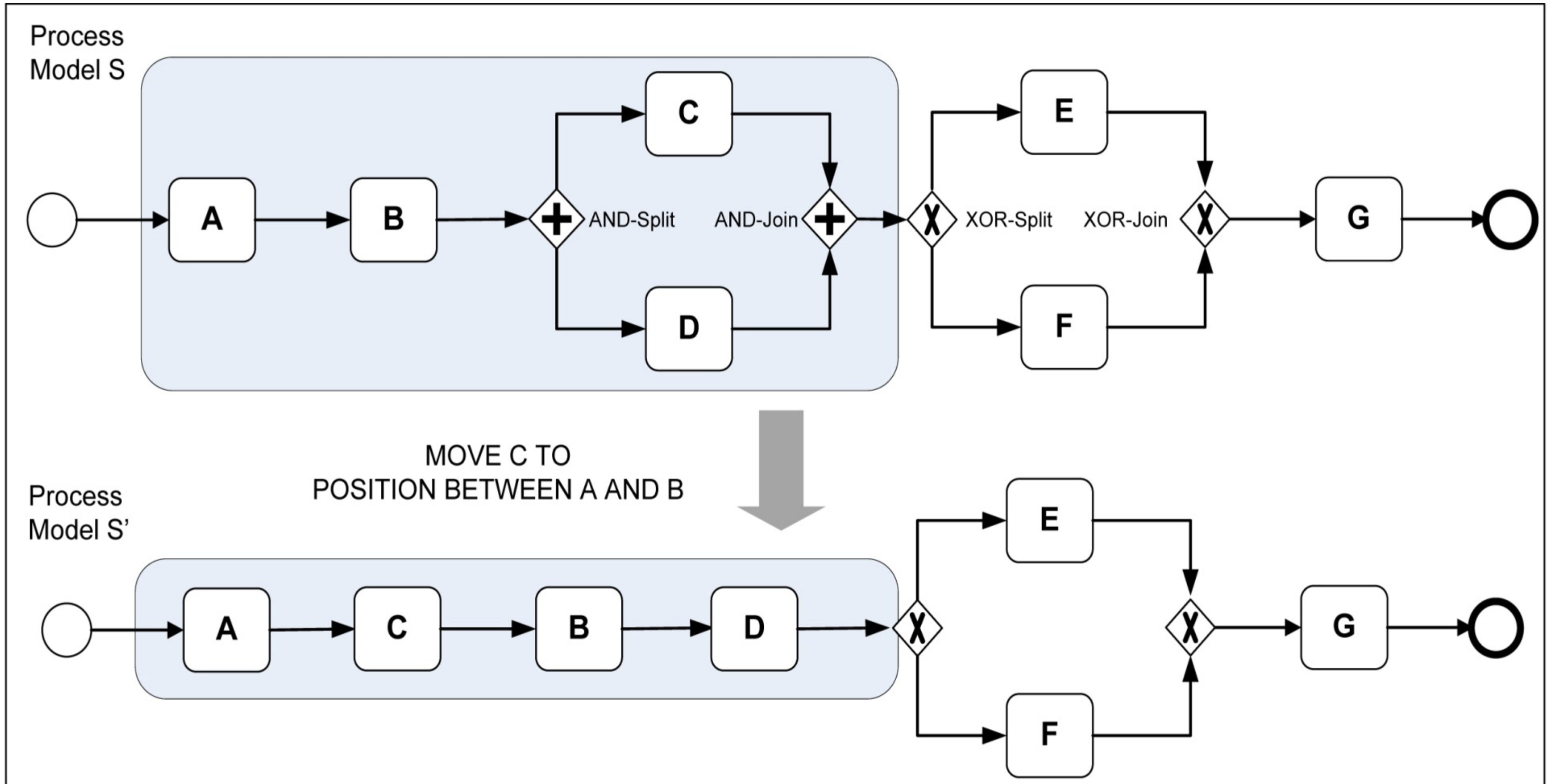
Manfred Reichert, Peter Dadam: ADEPTflex-Supporting Dynamic Changes of Workflows Without Losing Control. J. Intell. Inf. Syst. 10(2): 93-129 (1998)

Structural Adaptations of Pre-Specified Process Models

- Change Primitives
  - Add node
  - Remove node
  - Add edge
  - Remove edge
  - Move edge

- High-Level Change Operations
  - Combines a set of change primitives
  - Referred to as **Adaptation Patterns** in the following

Adaptation Patterns versus Change Primitives



Snapshot difference (expressed in terms of change primitives)

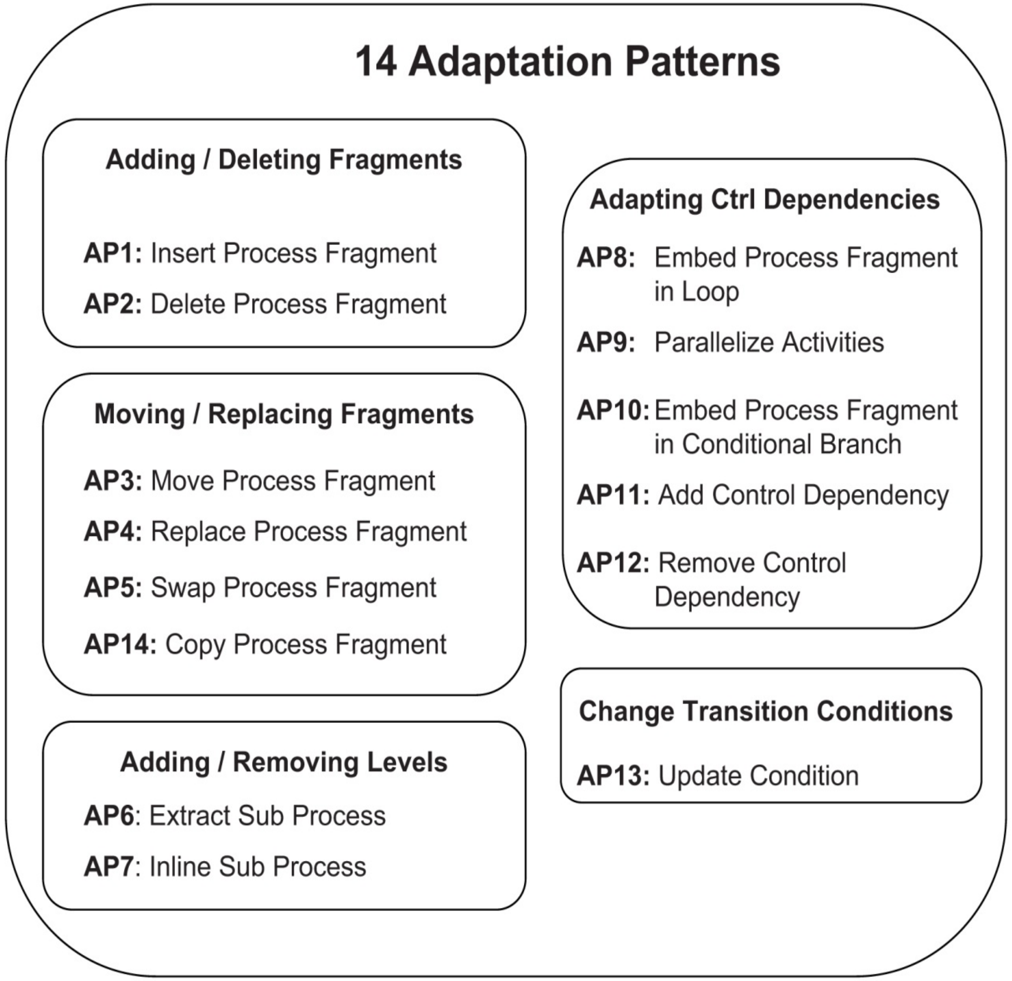
01: Delete edge from A to B
02: Delete edge from B to AND-Split
03: Delete edge from AND-Split to C
04: Delete edge from AND-Split to D
05: Delete edge from C to AND-Join
06: Delete edge from D to AND-Join
07: Delete edge from AND-Join to XOR-Split
08: Delete node AND-Split
09: Delete node AND-Join
10: Add edge from A to C
11: Add edge from C to B
12: Add edge from B to D
13: Add edge from D to XOR-Split

M. Reichert, B.Weber:
Enabling Flexibility in Process-Aware Information Systems,
@ Springer-Verlag Berlin Heidelberg 2012

Adaptation Patterns versus Change Primitives

Change Primitives	Process Adaptation Patterns
Operate on single elements of process schema	Provide high-level change operations
Correctness has to be checked after adaptation	Correctness-by-construction
No Assumption regarding structure of process schema	Process schema needs to be block-structured

Adaptation Patterns



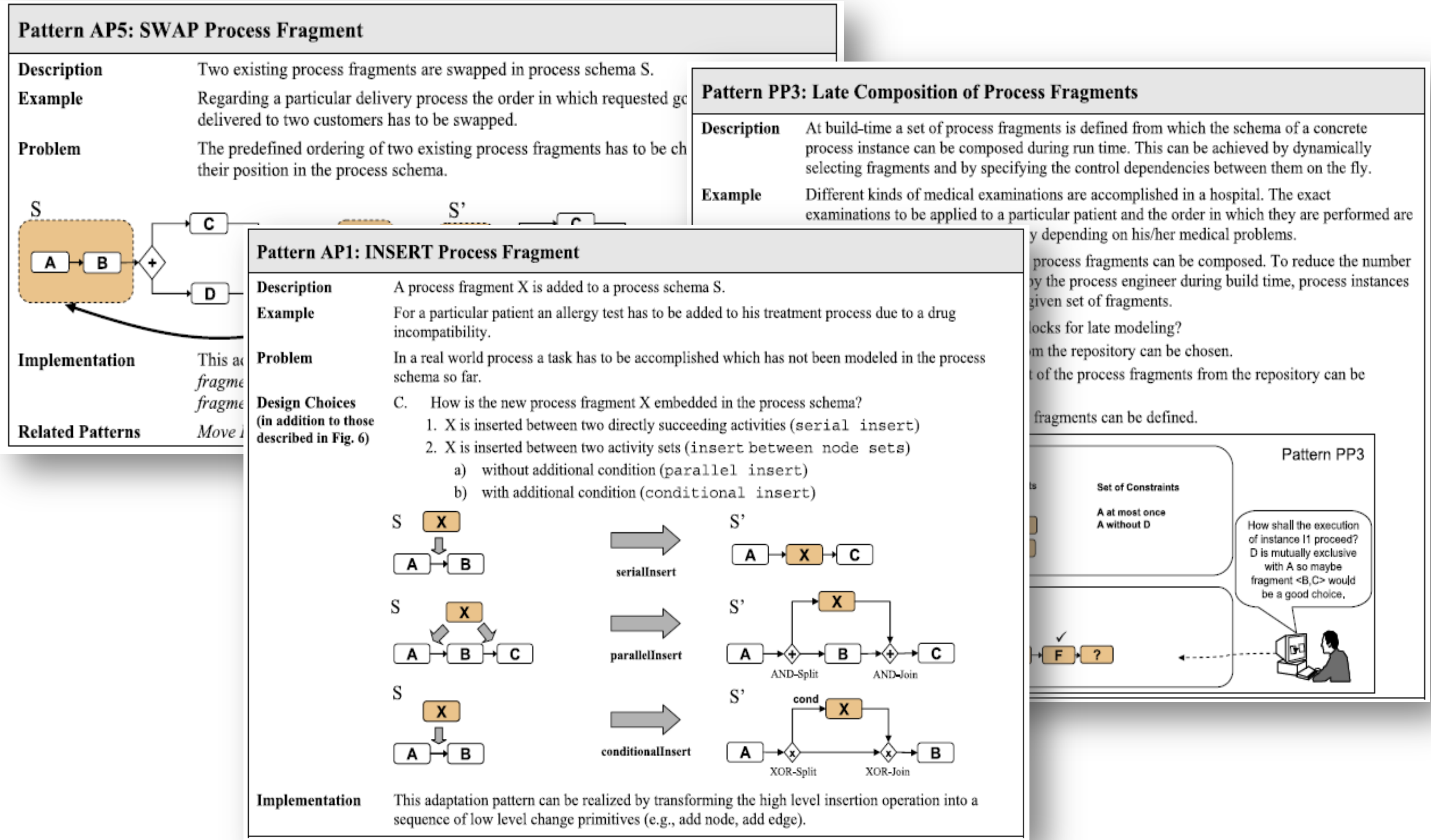
M. Reichert, B.Weber: Enabling Flexibility in Process-Aware Information Systems, @ Springer-Verlag Berlin Heidelberg 2012

ELSEVIER
Data & Knowledge Engineering
Volume 66, Issue 3, September 2008, Pages 438-466

Change patterns and change support features – Enhancing flexibility in process-aware information systems

Barbara Weber [a], Manfred Reichert [b, c], Stefanie Rinderle-Ma [b]

Catalogue of Adaptation Patterns


Pattern AP5: SWAP Process Fragment

Description Two existing process fragments are swapped in process schema S.
Example Regarding a particular delivery process the order in which requested go… delivered to two customers has to be swapped.
Problem The predefined ordering of two existing process fragments has to be ch… their position in the process schema.
Implementation This a… *fragme… fragme…*
Related Patterns *Move I…*

Pattern PP3: Late Composition of Process Fragments

Description At build-time a set of process fragments is defined from which the schema of a concrete process instance can be composed during run time. This can be achieved by dynamically selecting fragments and by specifying the control dependencies between them on the fly.
Example Different kinds of medical examinations are accomplished in a hospital. The exact examinations to be applied to a particular patient and the order in which they are performed are …y depending on his/her medical problems.
…process fragments can be composed. To reduce the number …y the process engineer during build time, process instances …given set of fragments.
…locks for late modeling?
…m the repository can be chosen.
…t of the process fragments from the repository can be
…fragments can be defined.

Pattern AP1: INSERT Process Fragment

Description A process fragment X is added to a process schema S.
Example For a particular patient an allergy test has to be added to his treatment process due to a drug incompatibility.
Problem In a real world process a task has to be accomplished which has not been modeled in the process schema so far.
Design Choices (in addition to those described in Fig. 6) C. How is the new process fragment X embedded in the process schema?
1. X is inserted between two directly succeeding activities (`serial insert`)
2. X is inserted between two activity sets (`insert between node sets`)
   a) without additional condition (`parallel insert`)
   b) with additional condition (`conditional insert`)

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

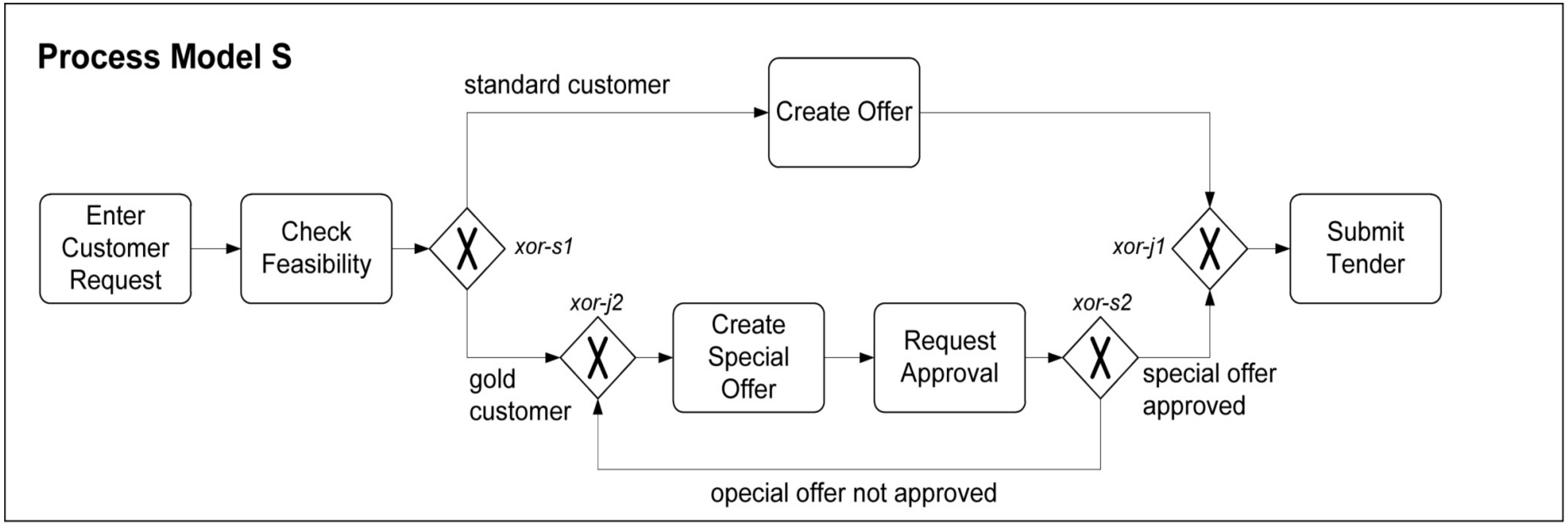
DTU

Process Evolution

- **Ability to change the implemented process when the real-world process changes**

- Immediateness of evolution
  - Deferred
    - Running instances not affected
  - Immediate
    - Running instances affected
    - Requires migration of instances

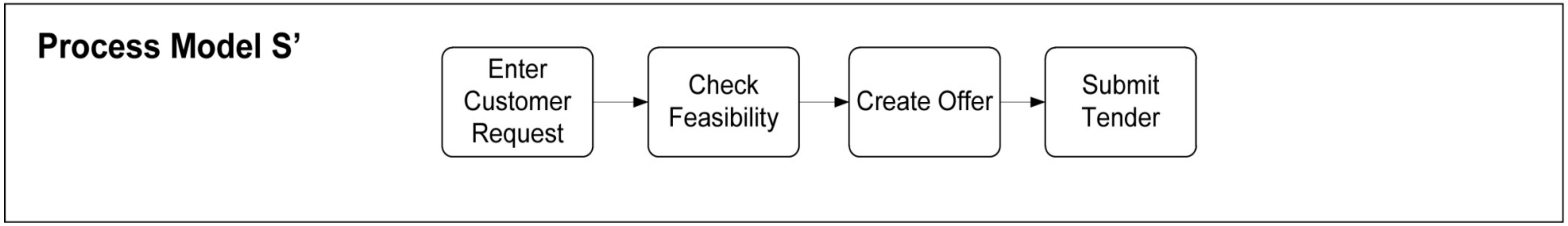
Process Schema Evolution



S evolves to S'

by applying model change τ_T with

τ_T = < **Move(S, Create Offer, Check Feasibility, xor-s1), Delete(S, (xor-s1,xor-j1)>**



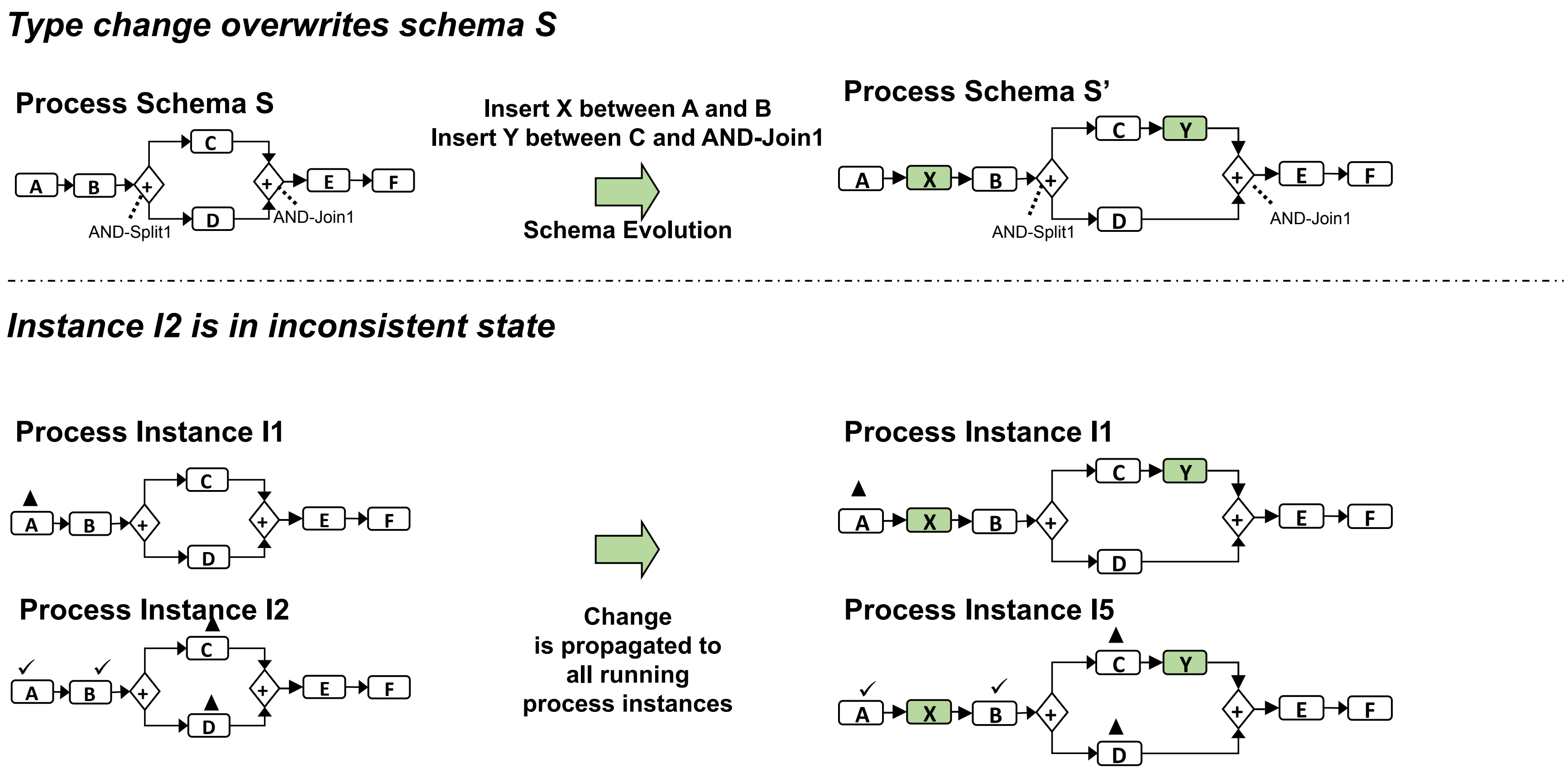
M. Reichert, B.Weber: Enabling Flexibility in Process-Aware Information Systems, @ Springer-Verlag Berlin Heidelberg 2012

Change Support Features
Schema Evolution, Version Control and Instance Migration

- Schema Evolution
  - Changes at the process type level
- How to deal with running instances when adapting the original process schema?
  - Scenario 1: No version control
  - Scenario 2: Co-existence of instances of old / new schema
  - Scenario 3: Change propagation and instance migration

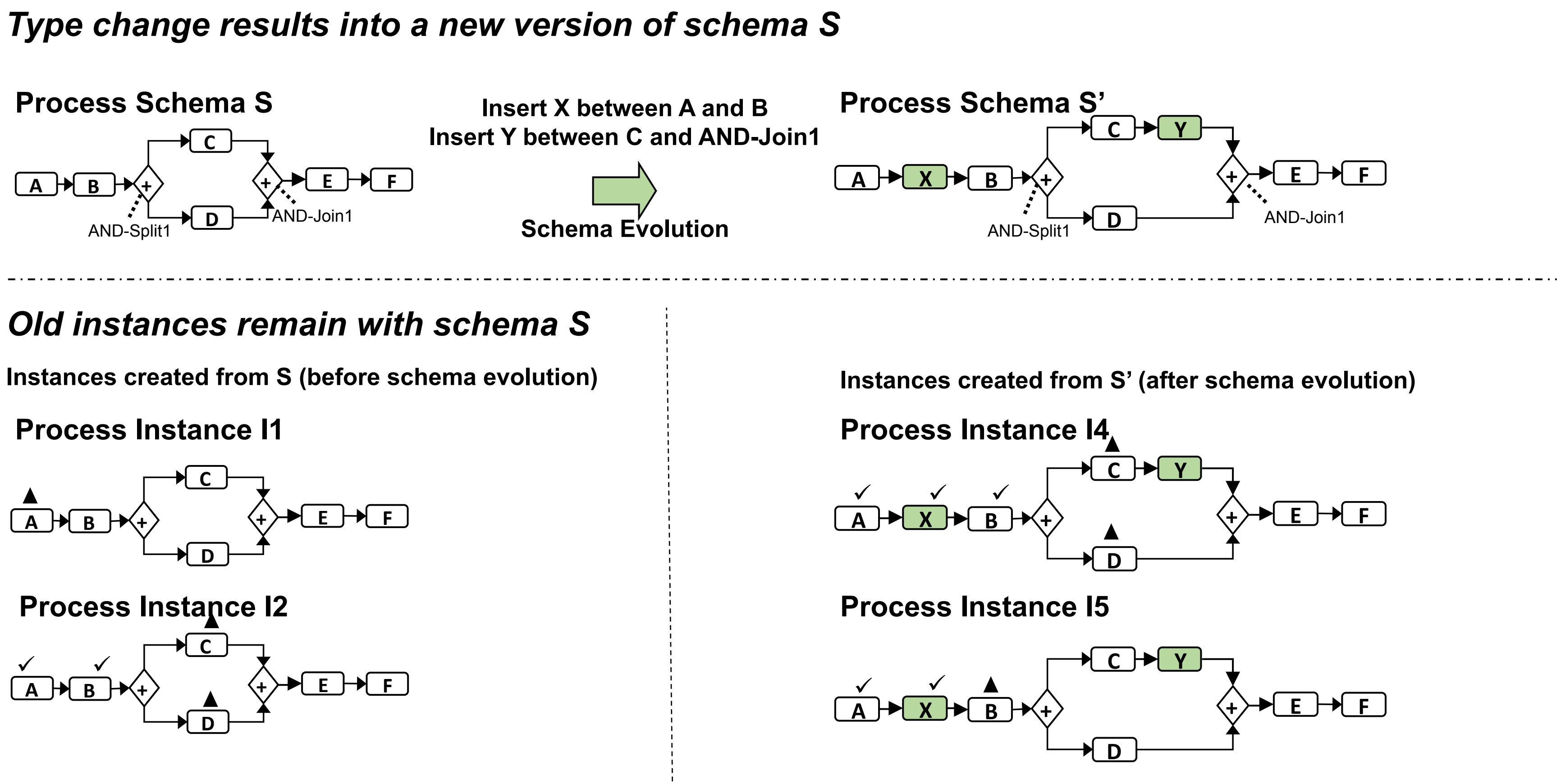
Scenario 1: No Schema Evolution

- Schema is overwritten and instances are migrated



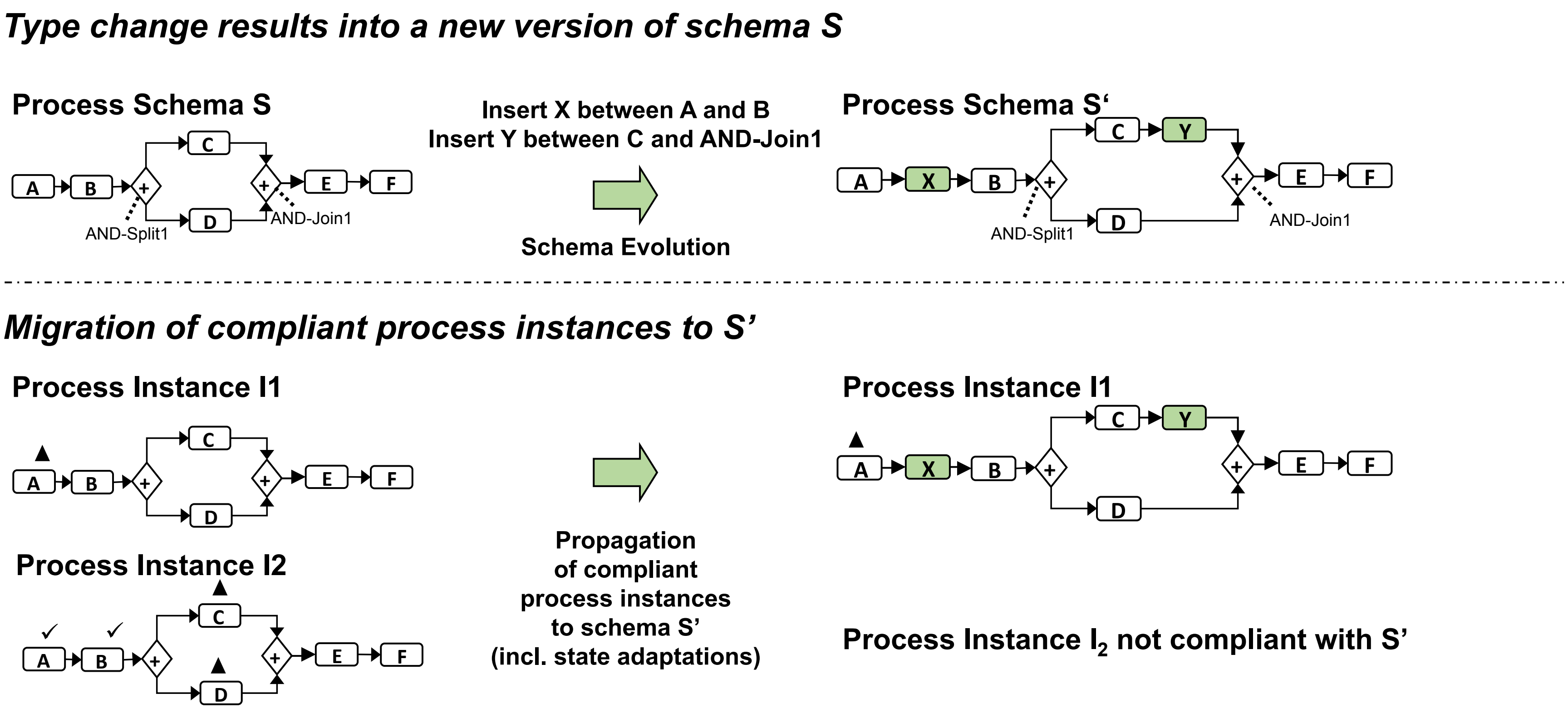
Scenario 2 – No version control

- Co-existence of instances of different schema versions



Scenario 3 – Instance Migration

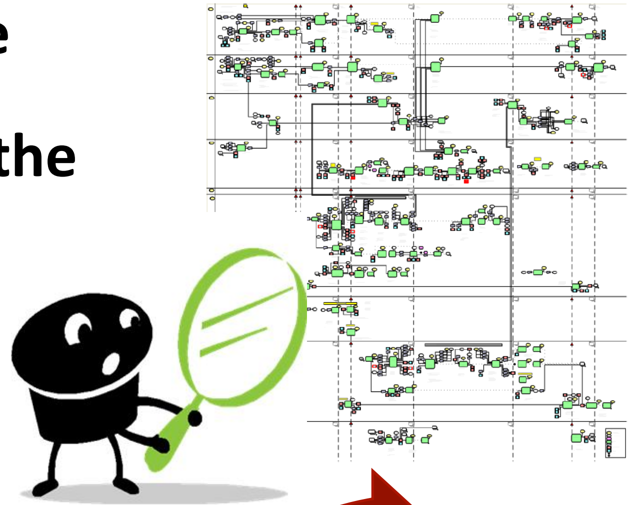
- Compliant instances are migrated to the new schema



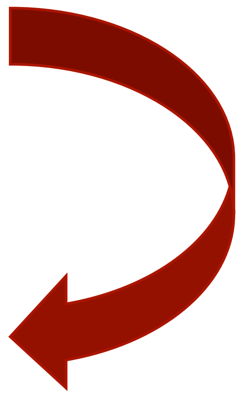
Stefanie Rinderle, Manfred Reichert, Peter Dadam: Correctness criteria for dynamic changes in workflow systems - a survey. Data Knowl. Eng. 50(1): 9-34 (2004)

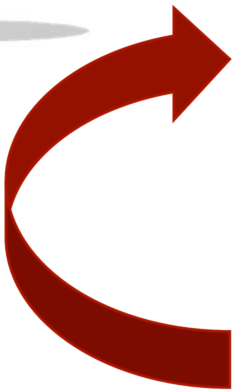
Process Model Refactoring

- **Improving model quality without changing the observable behavior of the model**



Identification of
Process Model Smells





Application of
Refactoring Techniques

Computers in Industry
Volume 62, Issue 5, June 2011, Pages 467-486

Survey paper

Refactoring large process model repositories

Barbara Weber [a], Manfred Reichert [b], Jan Mendling [c], Hajo A. Reijers [d]

Catalogue of Process Model Smells

Process Model Smells
PMS1: Non-intention revealing naming of activity / process model
PMS2: Contrived complexity
PMS3: Redundant Process Framgents
PMS4: Large Process Models
PMS5: Lazy Process Models
PMS6: Unused Branches
PMS7: Frequently Occuring Instance Changes
PMS8: Frequently Occurring Variant Changes

Catalogue of Refactoring Techniques

Process Model Refactoring
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
RF9: Generalize Variant Change
RF10: Remove Unused Branch
RF11: Pull Up Instance Change

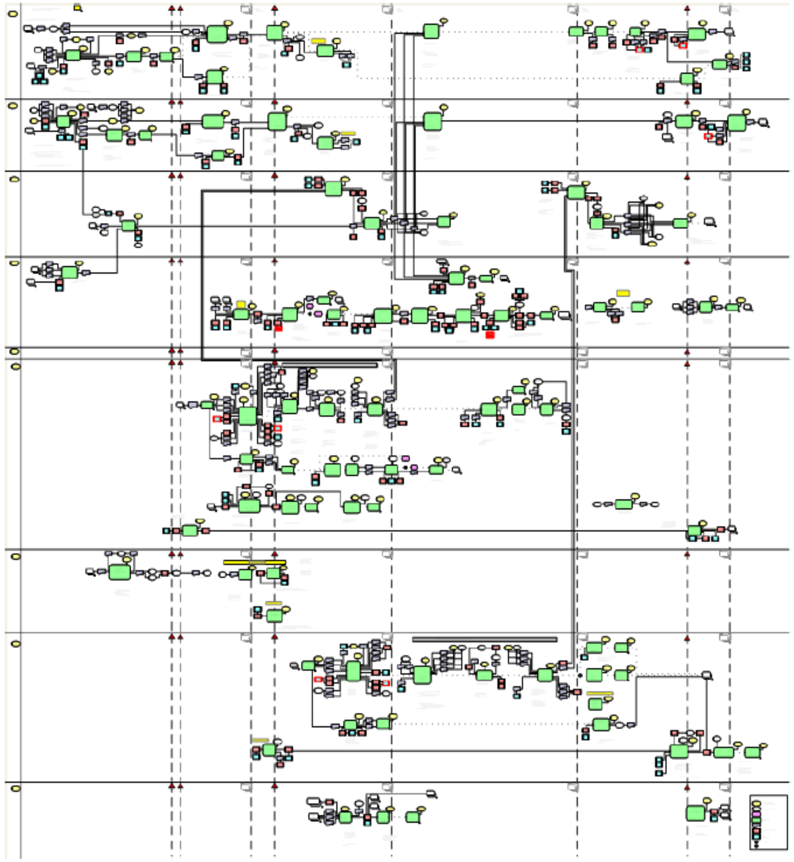
Labeling of Process Models (Example)

- **PMS1: Non intention revealing naming of activities / process models**
  - Ambiguous or non intention revealing labels
  - Inconsistent use of labels and labeling styles
- **Remedy: RF1: Rename activity**

- Example: Repository with 70 process models from healthcare
- 16 out of 70 process models contained activities regarding the scheduling of medical procedures (e.g., surgeries, medical examinations, drug administration)
- Although activities had similar intentions, different labels and labeling styles were used *"Make appointment", "appointment", "schedule examination", "fix day", "agree on surgery date", "plan"*

Large Process Model (Example)

- **PMS4: Large Process Model**
    - 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)
- **Remedy: RF4: Extract Process Fragment**

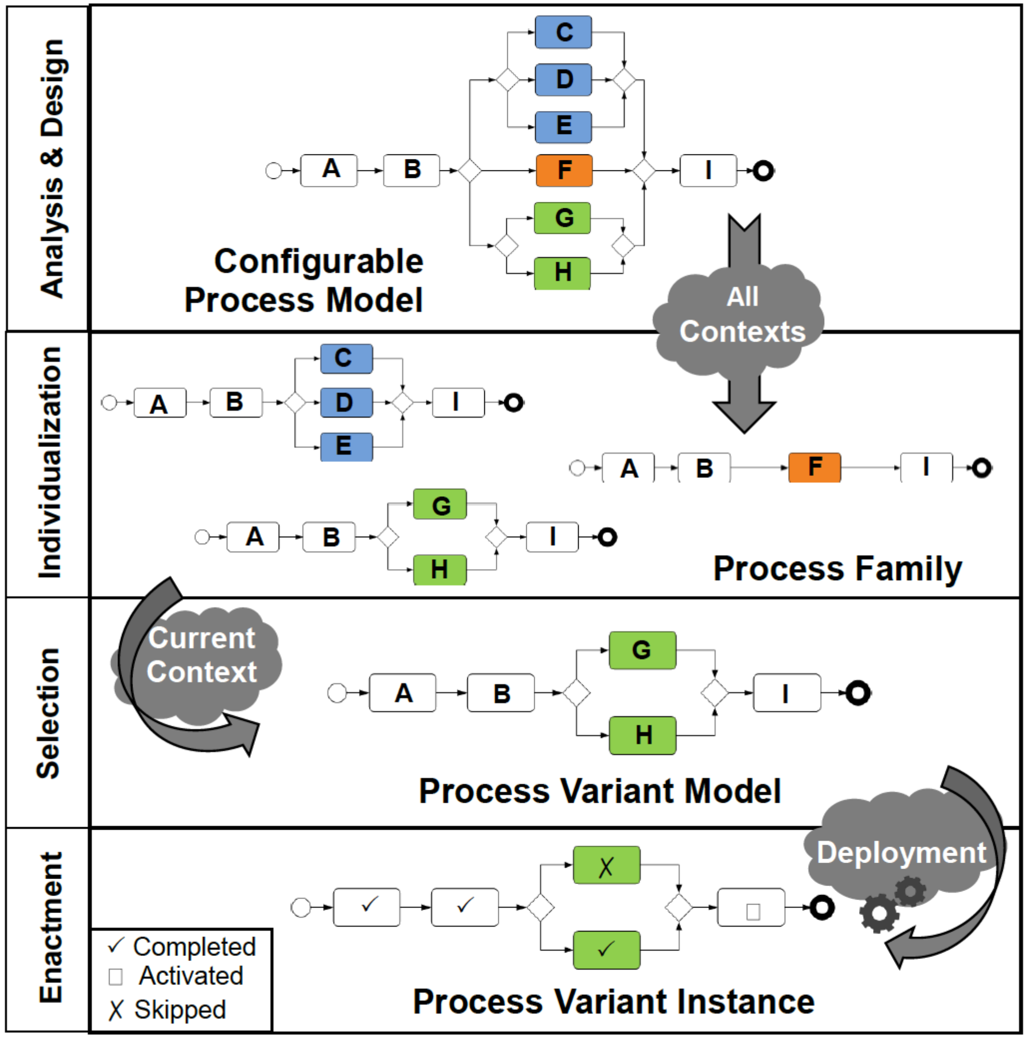


DTU

Business Process Variability

- **Variability requires that processes, depending on the context, are treated differently**
- Context Factors are known and selection of specific variant depends on context
- Typical Driver
  - Product and Service Variability
  - Country-specific (legal) regulations
  - Different customer groups
  - Seasonal differences

From Process Family Definition to Variant Enactment



Information and Software Technology
Volume 57, January 2015, Pages 248-276

VIVACE: A framework for the systematic evaluation of variability support in process-aware information systems

Clara Ayora [a], Victoria Torres [a, 1], Barbara Weber [b, 2], Manfred Reichert [c, 3], Vicente Pelechano [a, 1]

VIVACE Framework

The *VIVACE* framework	
Modeling language used to represent process variability	
Technique used for building the configurable process model	
Method for modeling the process family	
Process perspectives covered	
Variability-specific language constructs	LC1 Configurable Region
	LC2 Configuration Alternative
	LC3 Configuration Context Condition
	LC4 Configuration Constraint
	LC5 Configurable Region Resolution Time
Variability support features	**Analysis & Design phase**
	F1.1 Modeling a configurable process model
	F1.2 Verifying a configurable process model and its related process family
	F1.3 Validating a configurable process model
	F1.4 Evaluating the similarity of different process variants
	F1.5 Merging process variants
	Configuration phase
	F2 Configuring specific regions of a process variant out of a configurable process model
	Enactment phase
	F3.1 Configuring specific regions of a process variant at enactment time
	F3.2 Dynamically re-configuring an instance of a process variant at enactment time
	Diagnosis
	F4 Analyzing a collection of process variants
	Evolution
	F5.1 Versioning of a configurable process model
	F5.2 Propagating changes of a configurable process model to already configured process variants
Tool implementation	
Empirical evaluation	
Application domain	

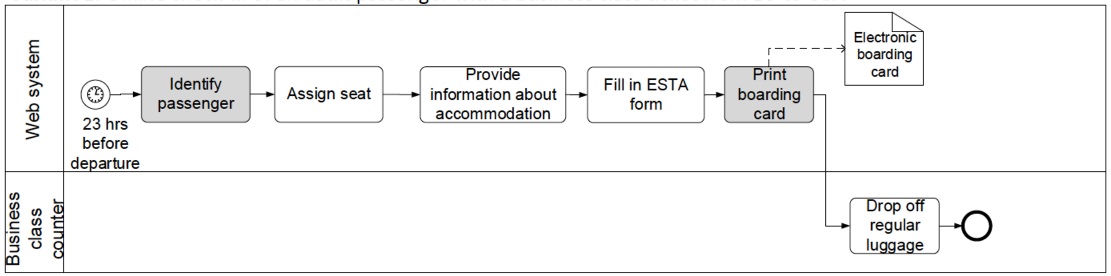
Outcome of a Systematic Literature Review

VIVACE Framework

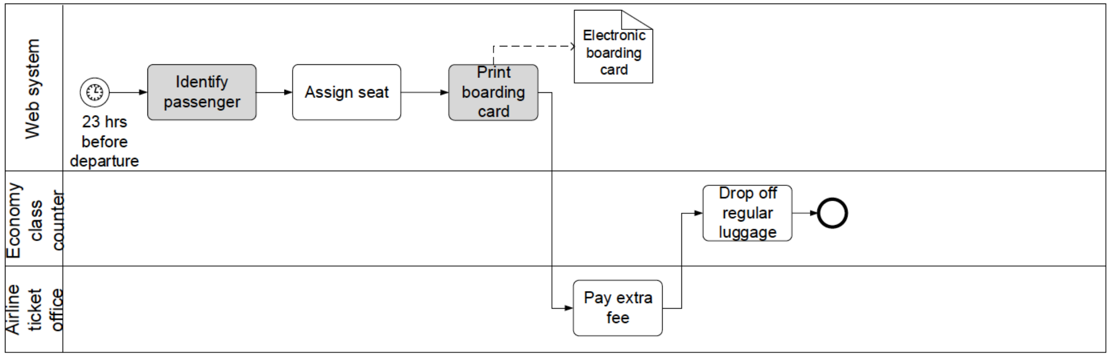
- **Variability-specific Language Constructs**
  - Configurable Region
  - Configuration Alternative
  - Configuration Context Condition
  - Configuration Constraint
  - Configurable Region Resolution Time

Variants of the Check-in Process (1)

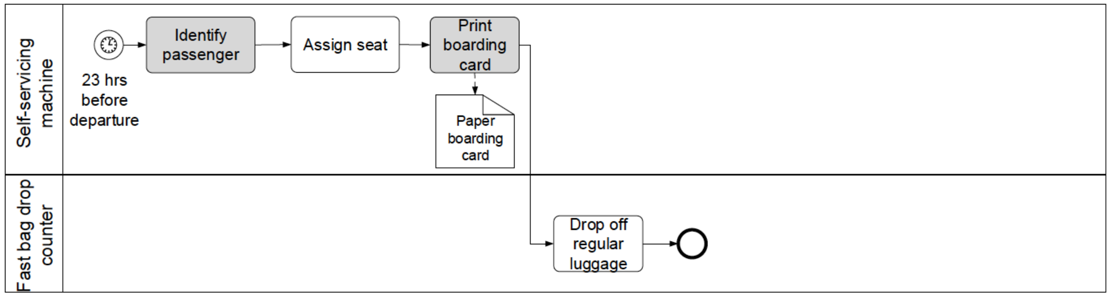
Variant 1: Online check-in of an adult passenger with a business class ticket from EU to USA



Variant 2: Online check-in of an adult passenger with an economy class ticket from EU to EU with overweight luggage

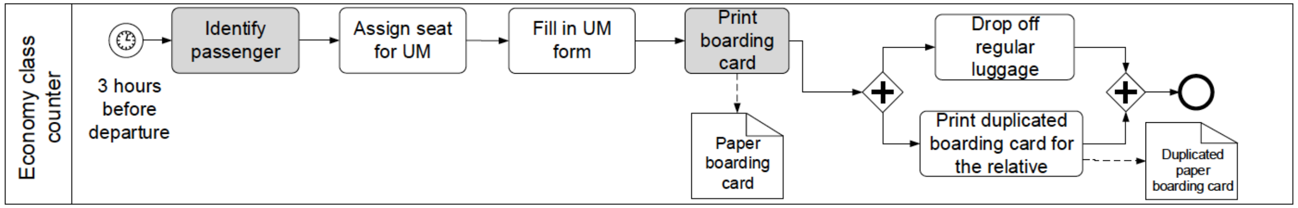


Variant 3: Check-in at the self-servicing machine for an adult passenger with an economy class ticket from EU to EU

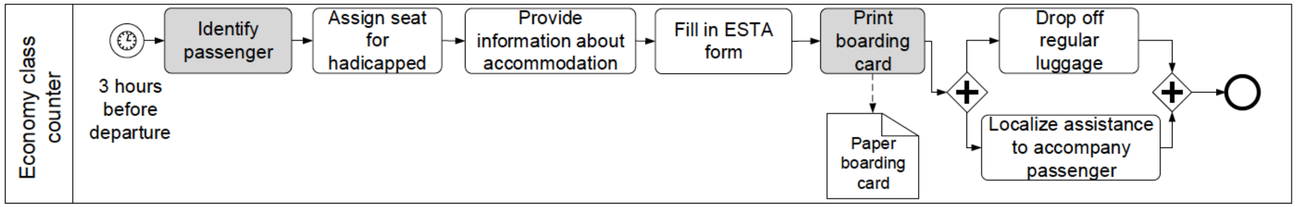


Variants of the Check-in Process (2)

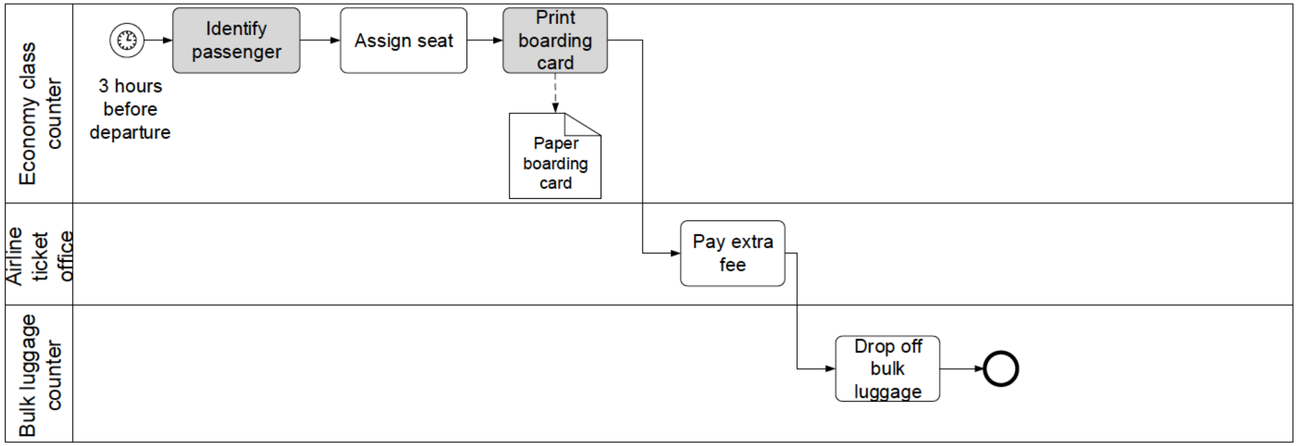
Variant 4: Check-in for an unaccompanied minor (UM) passenger with an economy class ticket from EU to EU with a relative accompanying him until the boarding gate



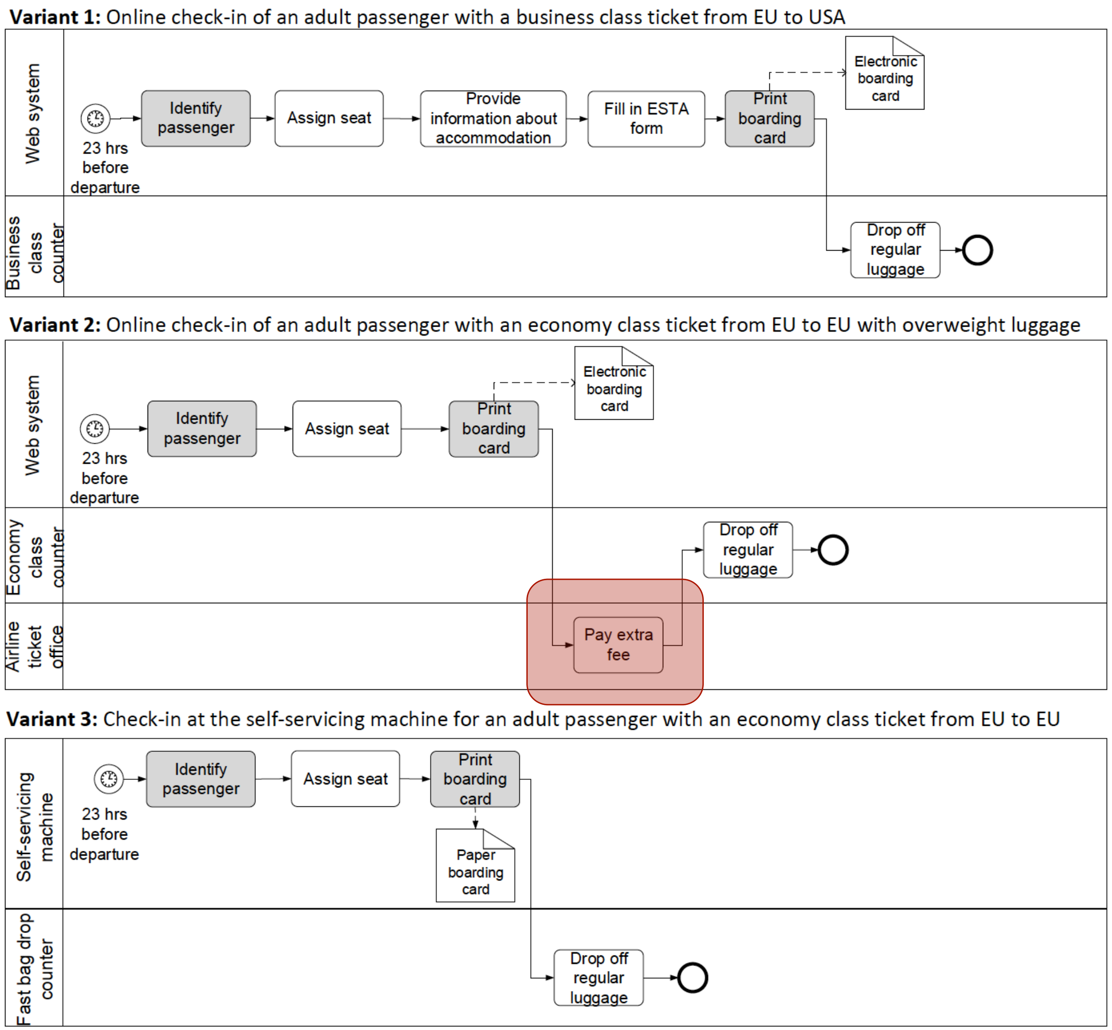
Variant 5: Check-in for a handicapped passenger with an economy class ticket from EU to USA



Variant 6: Check-in for an adult passenger with an economy class ticket from EU to EU with bulk luggage

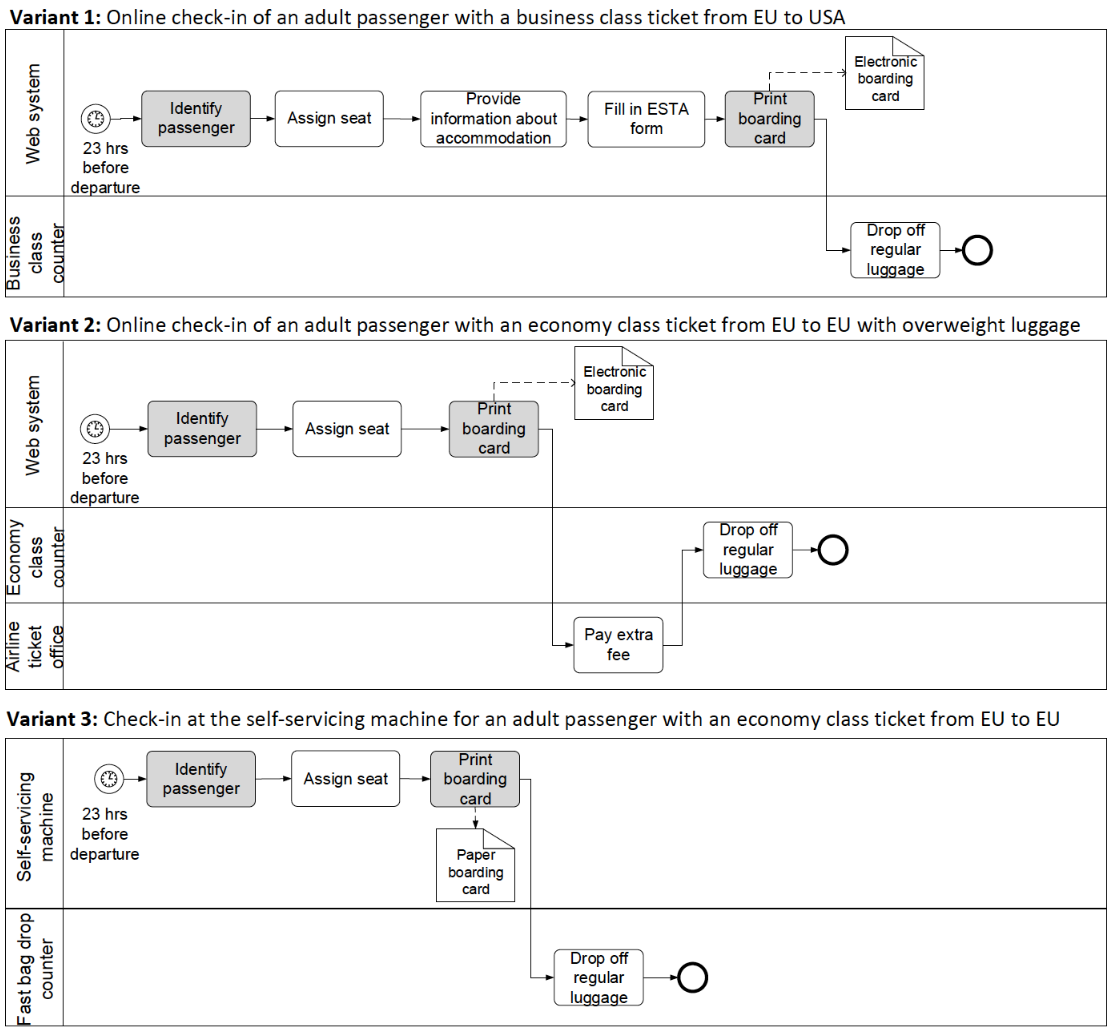


Configurable Region



- Activity *Pay extra fee* is only performed if the luggage has overweight. Otherwise, it is skipped.
- Modeling this variability requires a **configurable region** in the configurable process model

Configuration Alternatives

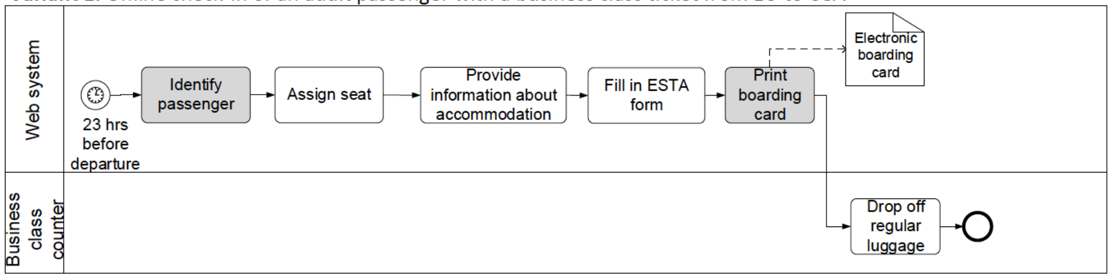


- Activity *Pay extra fee* is only performed if the luggage has overweight. Otherwise, it is skipped

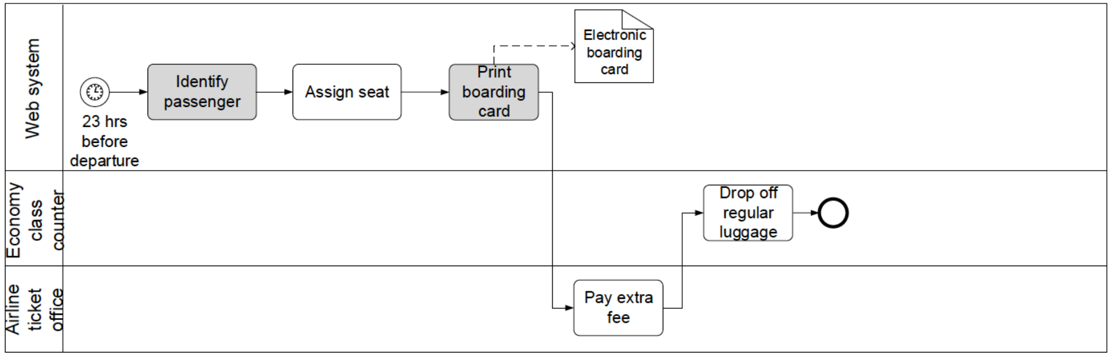
- Two **configuration alternatives**: either perform the activity *Pay extra fee* or skip it

Configuration Context Condition

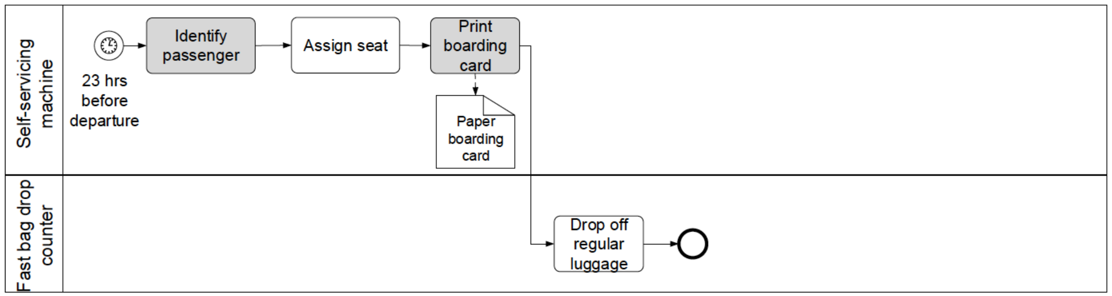
Variant 1: Online check-in of an adult passenger with a business class ticket from EU to USA



Variant 2: Online check-in of an adult passenger with an economy class ticket from EU to EU with overweight luggage

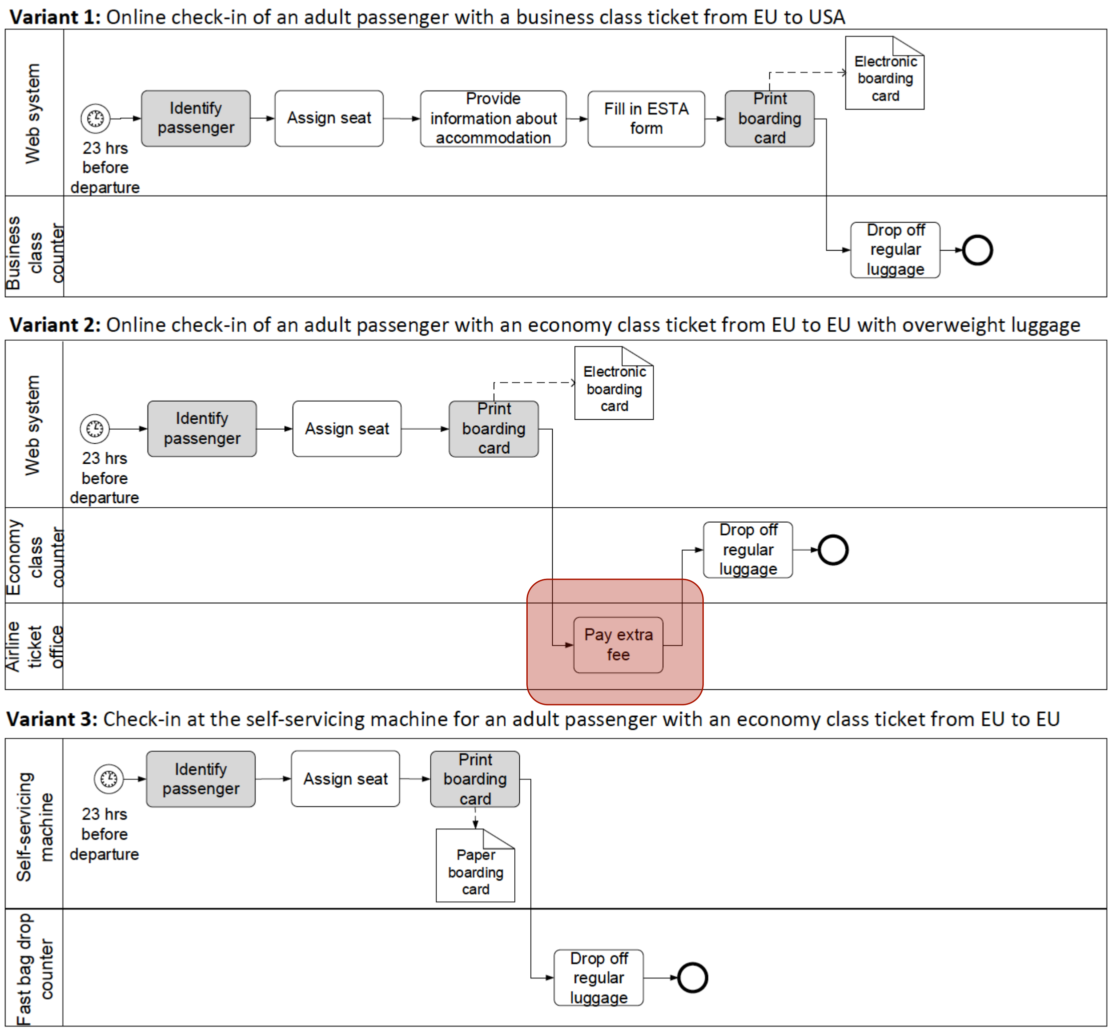


Variant 3: Check-in at the self-servicing machine for an adult passenger with an economy class ticket from EU to EU



- Activity *Pay extra fee* is only performed if the luggage has overweight. Otherwise, it is skipped
- **Context condition**: luggage has overweight

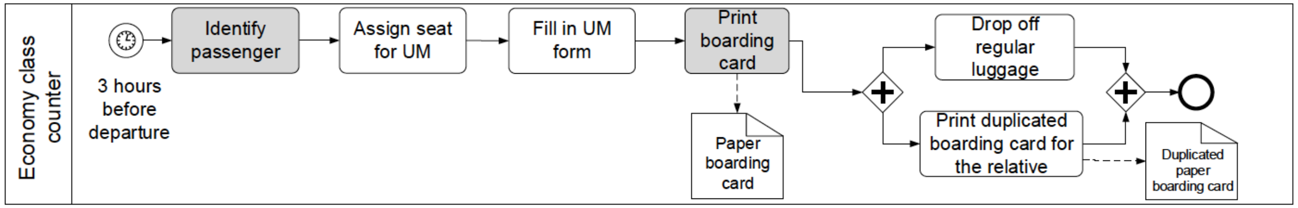
Configurable Region Resolution Time



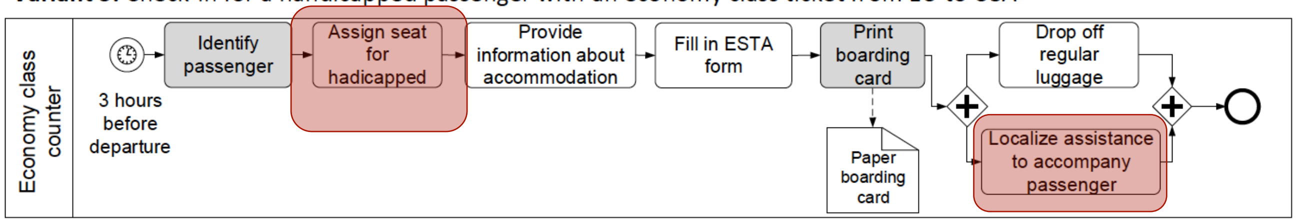
- Activity *Pay extra fee* is only performed if the luggage has overweight. Otherwise, it is skipped
- **Resolution time**: it often only becomes clear during check-in (i.e., at run-time) whether or not an extra fee needs to be paid

Configuration Constraint

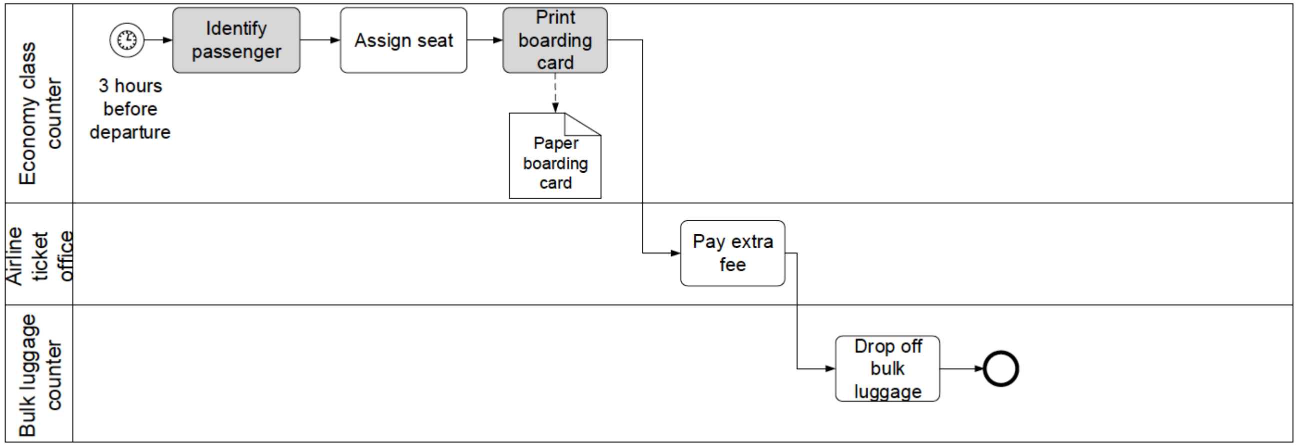
Variant 4: Check-in for an unaccompanied minor (UM) passenger with an economy class ticket from EU to EU with a relative accompanying him until the boarding gate



Variant 5: Check-in for a handicapped passenger with an economy class ticket from EU to USA



Variant 6: Check-in for an adult passenger with an economy class ticket from EU to EU with bulk luggage



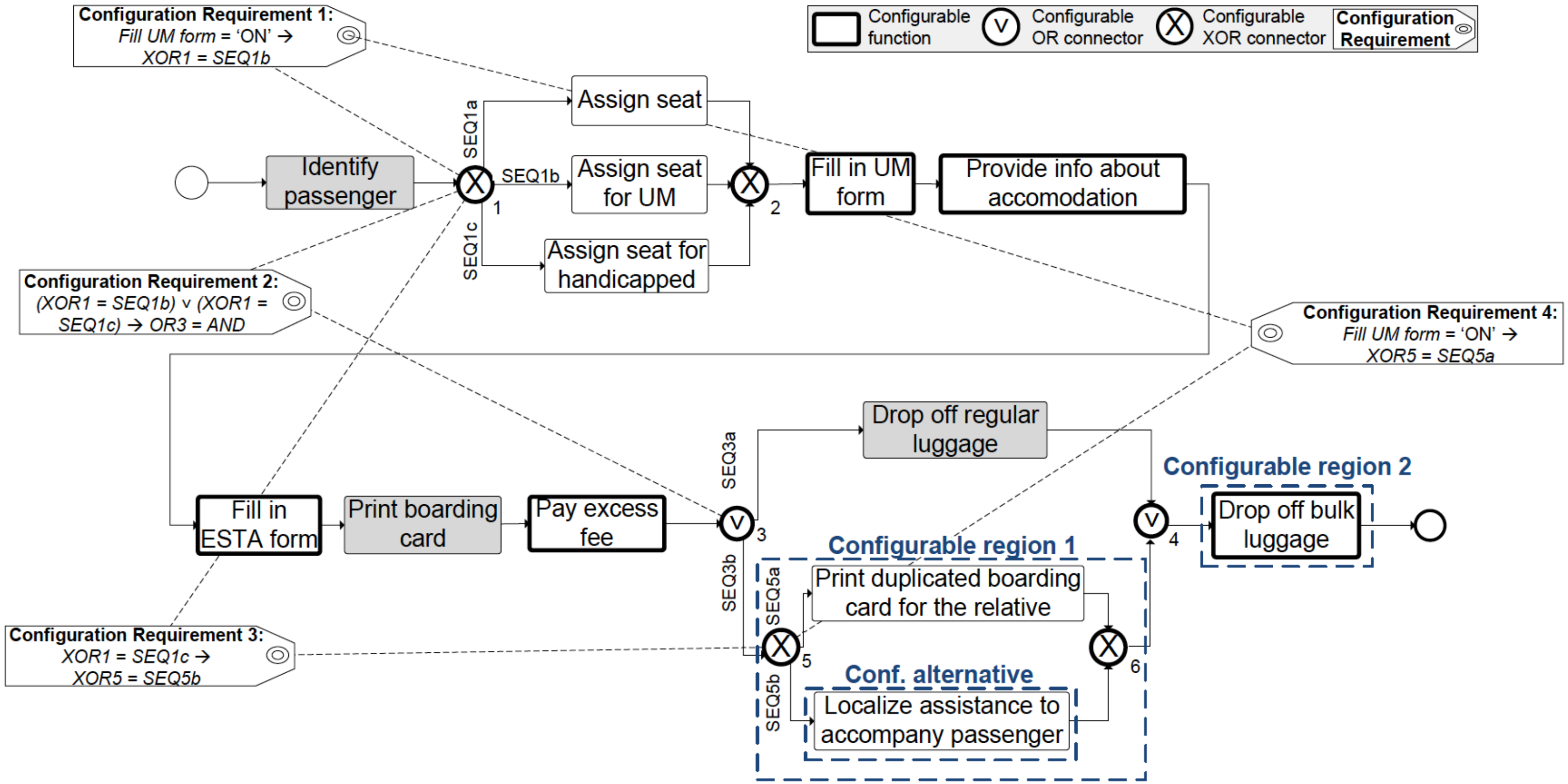
- *Localize assistance to accompany passenger* should only be performed for passengers with handicap, i.e., there exists a **configuration constraint**

Assessing Different Approaches to BP Variability using the VIVACE Framework

- VIVACE Framework can be used to assess different approaches to support variability in business processes
- Two Main approaches
  - Behavioural-based approaches
  - Structural-based approaches

Behavioural-based Approaches

- Example: C-EPC/C-iEPC

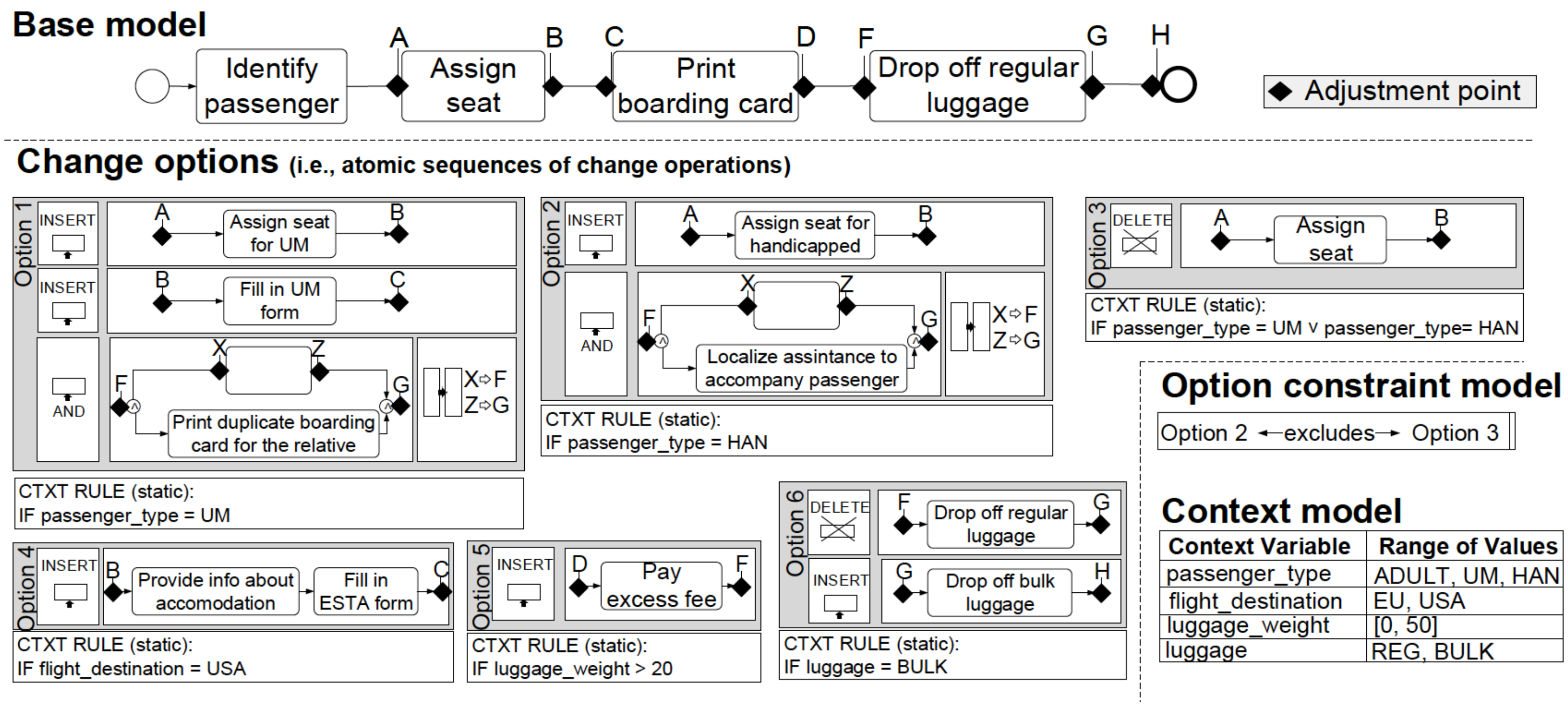


Florian Gottschalk, Wil M. P. van der Aalst, Monique H. Jansen-Vullers, Marcello La Rosa: Configurable Workflow Models. Int. J. Cooperative Inf. Syst. 17(2): 177-221 (2008)

Michael Rosemann, Wil M. P. van der Aalst: A configurable reference modelling language. Inf. Syst. 32(1): 1-23 (2007)

Structural-based Approaches

- Example: Provop



Alena Hallerbach, Thomas Bauer, Manfred Reichert: Capturing variability in business process models: the Provop approach. Journal of Software Maintenance 22(6-7): 519-546 (2010)

The Process Spectrum

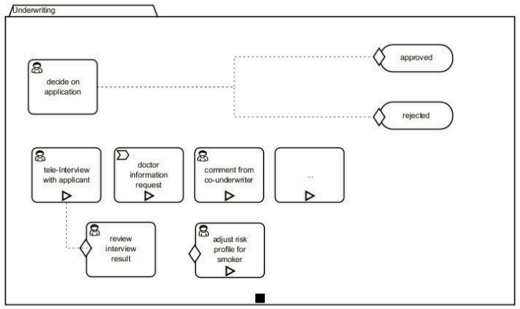
- The process spectrum reaches from
  - completely predictable and highly repetitive
  - to completely unpredictable and little repetitive

Not today's focus

EcoKnow project

Innovation Fund Denmark
RESEARCH, TECHNOLOGY & GROWTH

Loosely-specified process models, e.g., CMNN, DCR Graphs



Specific property: **Emergence**

Completely unpredictable
Highly repetitive

Completely unpredictable
Little repetitive

Nautilus
(FWF P23699-N23

Modeling Mind
(FWF P26609-N15)

ModErARe
(FWF P26140-N15)

Investigating the Process of Process Modeling

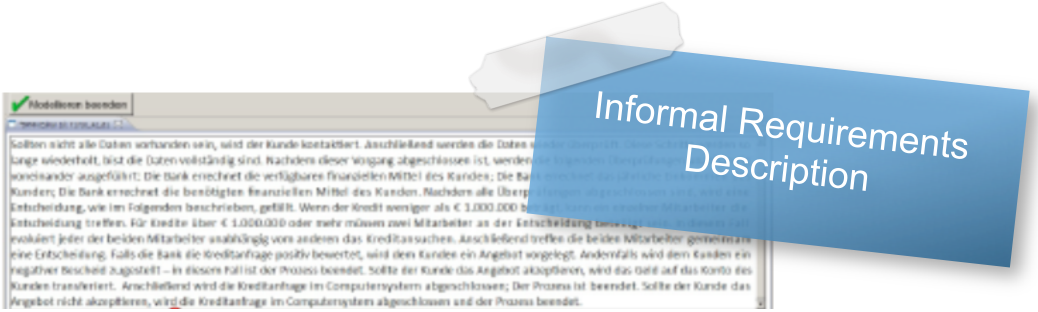
DTU Compute
Department of Applied Mathematics and Computer Science

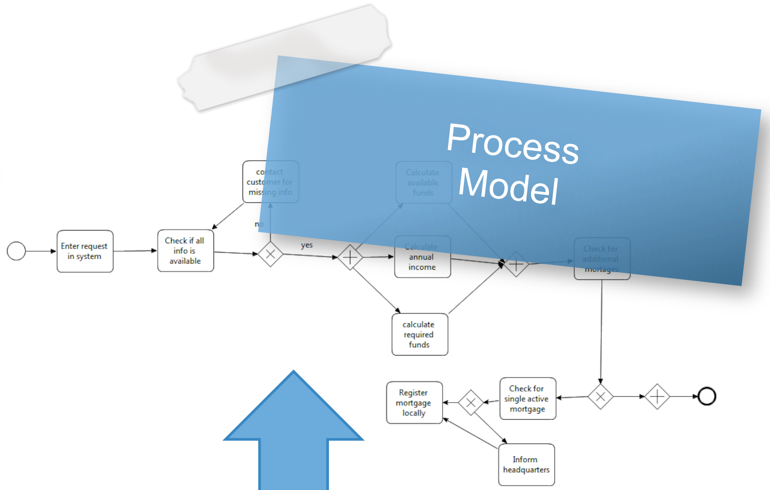
Process Model Creation
An Example of an Interactive Design Activity

- Process model creation can be characterized as an interactive design activity

Process Model Creation
An Example of an Interactive Design Activity

- Process model creation can be characterized as an interactive design activity

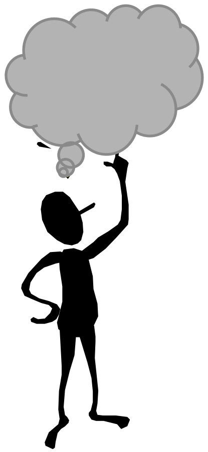




Process Model Creation as Interactive Design Activity

Process Model Creation
An Example of an Interactive Design Activity

- Process model creation can be characterized as an interactive design activity



1 Mental model

Internal representation

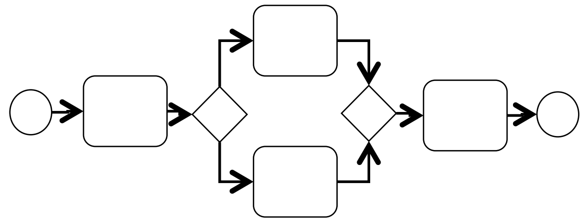
Process Model Creation
An Example of an Interactive Design Activity

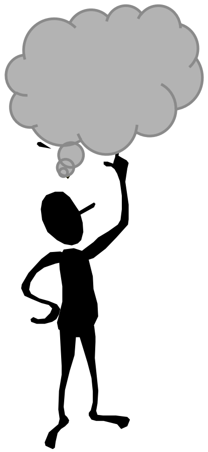
- Process model creation can be characterized as an interactive design activity

2

Physical model

External representation





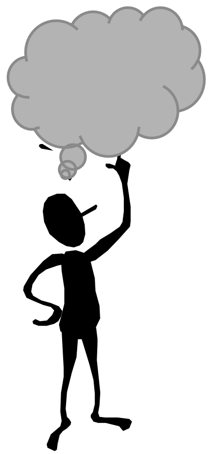
1

Mental model

Internal representation

Process Model Creation
An Example of an Interactive Design Activity

- Process model creation can be characterized as an interactive design activity

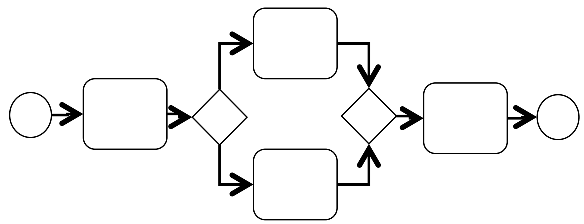


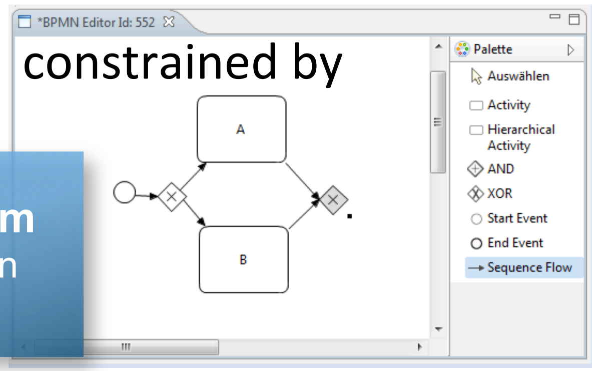
1 **Mental model**

Internal representation

2 **Physical model**

External representation

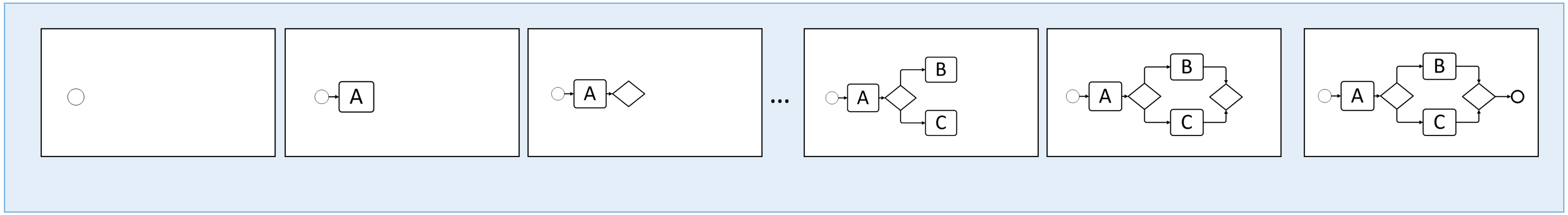




Modeling Platform
- Modeling notation
- Tool support

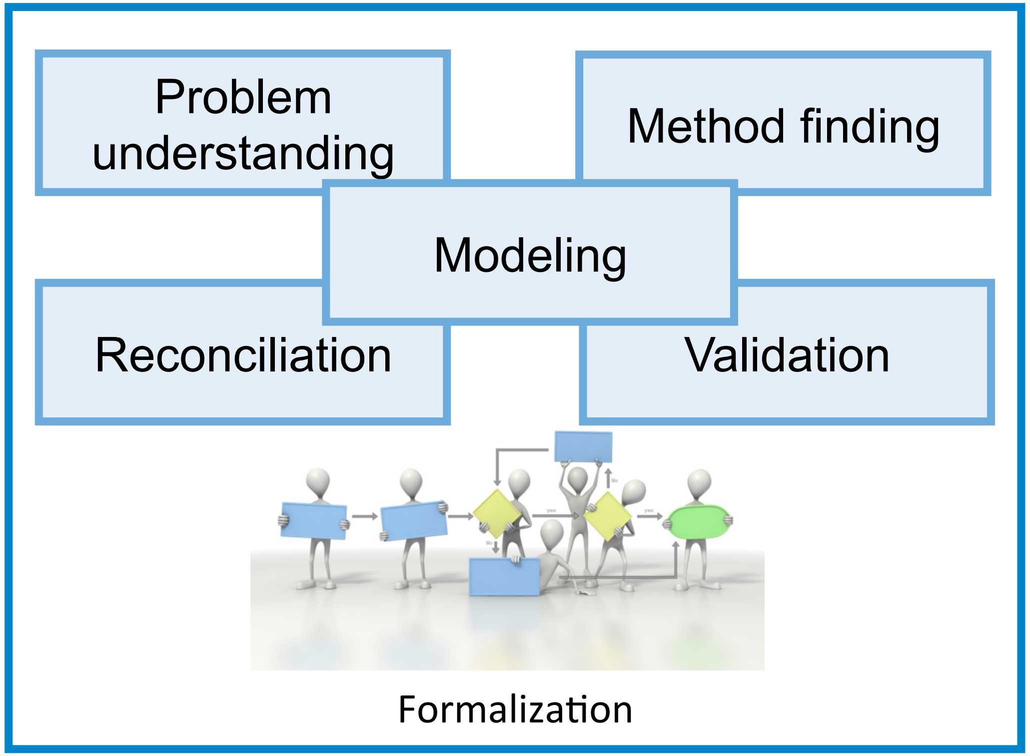
Pnina Soffer, Maya Kaner, Yair Wand: Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. Business Process Management Workshops (1) 2011: 357-369

Process Model Creation
An Example of an Interactive Design Activity



- Modeler evolves the design artifact
- Design artifact can be characterized by a set of properties
  - E.g., number of line crossings, orthogonality of edges, etc.
- Properties of design artifact change as the artifact evolves

Process Model Creation
Phases of Process Model Creation

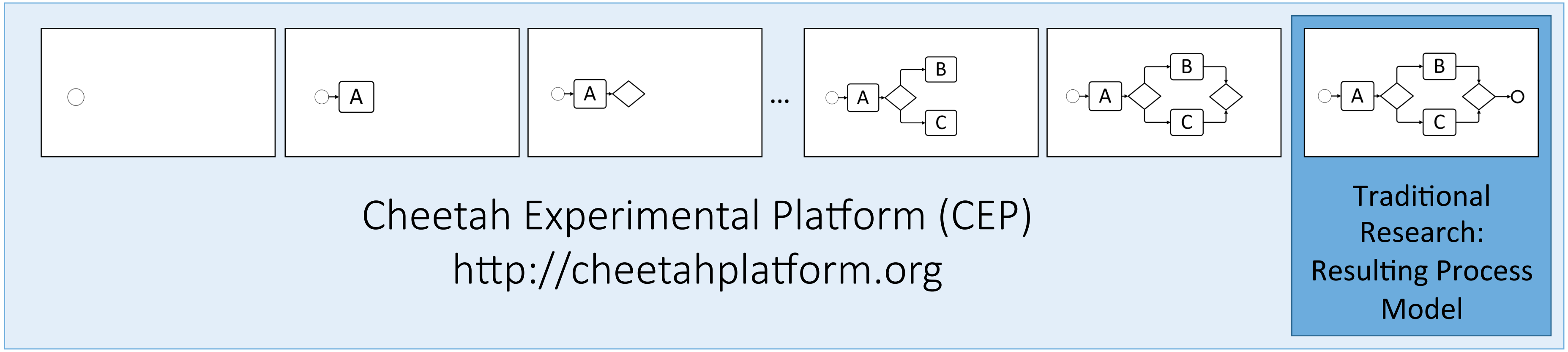


J. Pinggera: The Process of Process Modeling. PhD thesis, University of Innsbruck, Department of Computer Science, 2014.

- Decomposed into different phases
- Iterative, highly flexible process

Investigating the Process of Process Modeling
Cheetah Experimental Platform

- Logging interactions with modeling platform
  - Model interactions
  - Technology use



J. Pinggera, S. Zugal and B. Weber: Investigating the Process of Process Modeling with Cheetah Experimental Platform. In: Proc. ER-POIS '10, pp. 13–18, 2010.

Investigating the Process of Process Modeling
Cheetah Experimental Platform

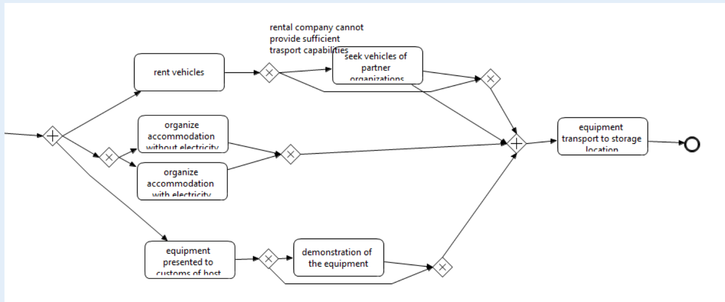
Conducting modeling sessions with Cheetah Experimental Platform (CEP)

PPM Instance

13	08:43:27	Horizontal Scroll
14	08:43:27	Horizontal Scroll
15	08:44:02	Create Activity 'transport to disaster area'
16	08:44:04	Move Activity 'transport to disaster area'
17	08:44:13	Rename Activity 'transport to disaster area' to 'transpo...
18	08:44:38	Create Activity 'present equipment'
19	08:44:43	Create Sequence Flow from Start Event to Activity 'tra...
20	08:44:47	Create Sequence Flow from Activity 'transport equip...
21	08:45:03	Create XOR
22	08:45:03	Vertical Scroll
23	08:45:16	Create Activity 'demonstrate equipment'
24	08:45:16	Vertical Scroll
25	08:45:16	Horizontal Scroll
26	08:45:18	Move Activity 'demonstrate equipment'
27	08:45:19	Horizontal Scroll

Logging of all model interactions
Basis for mining a modeler's behavior

Final Process Model



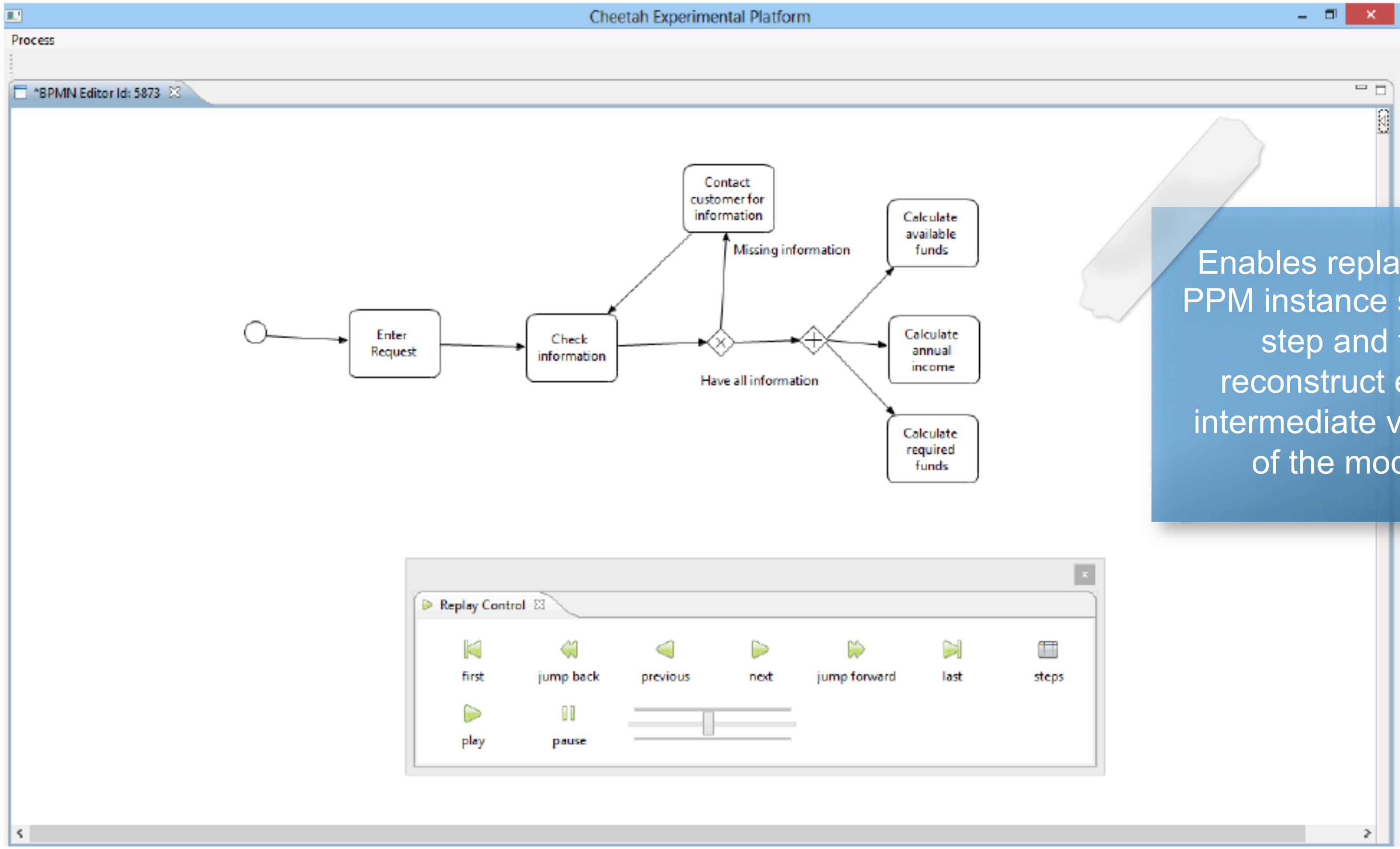
J. Pinggera, S. Zugal and B. Weber: Investigating the Process of Process Modeling with Cheetah Experimental Platform. In: Proc. ER-POIS '10, pp. 13–18, 2010.

Investigating the Process of Process Modeling
Interactions Logged by Cheetah Experimental Platform

Type of Modeler Interaction	Description
CREATE NODE	Create activity or gateway
CREATE EDGE	Create an edge connecting two nodes
CREATE CONDITION	Create an edge condition
RECONNECT EDGE	Reconnect an edge fron one node to another
DELETE NODE	Delete activity or gateway
DELETE EDGE	Delete an edge conneting two nodes
DELETE CONDITION	Delete an edge condition
RENAME	Rename an activity
MOVE NODE	Move a node
MOVE EDGE LABEL	Move the label of an edge
CREATE/DELETE/MOVE EDGE BENDPOINT	Update the routing of an edge
UPDATE CONDITION	Update an edge's condition
VSCROLL	Scroll vertically
HSCROLL	Scroll horizontally

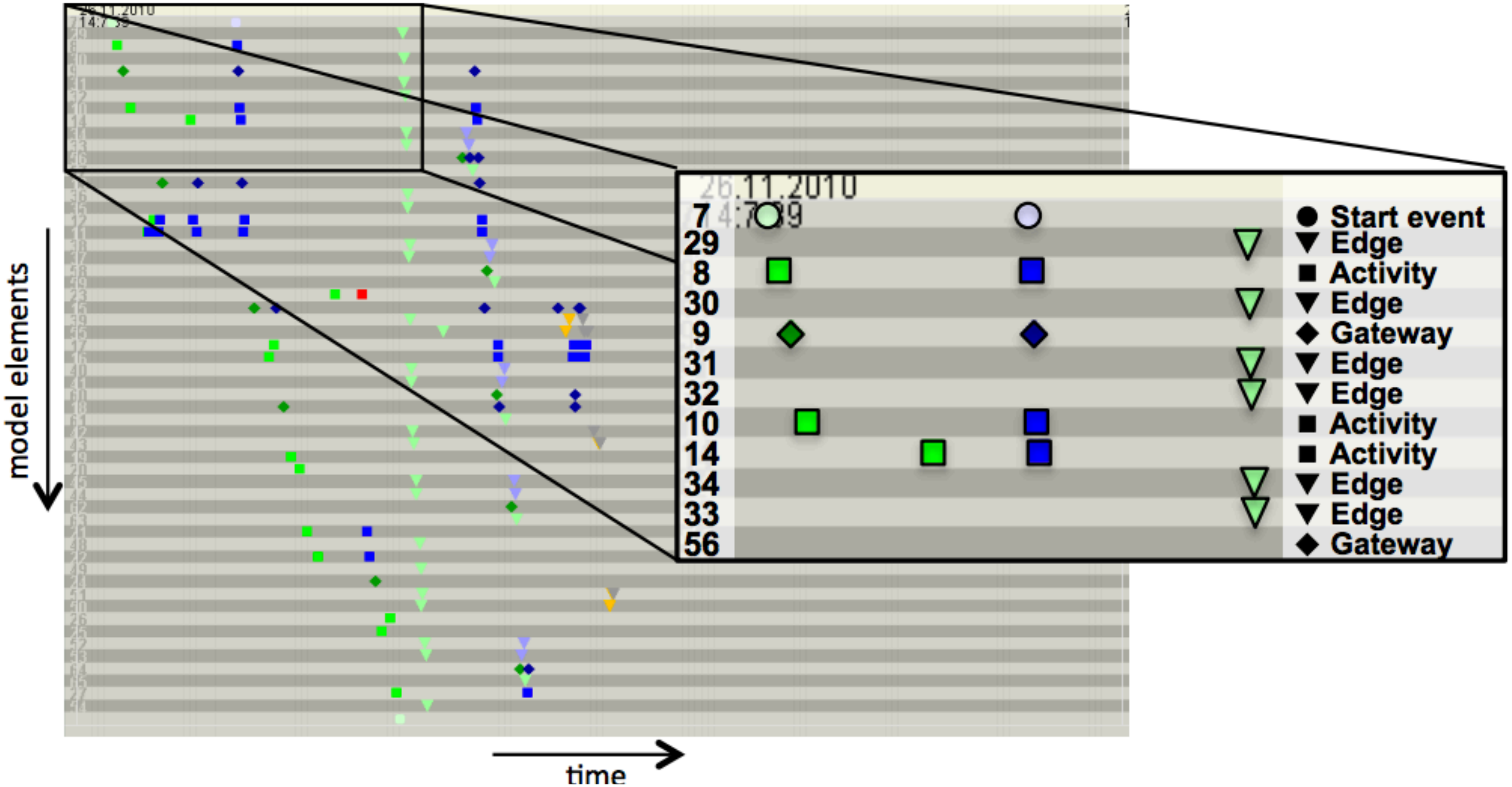
Investigating the Process of Process Modeling
Cheetah Experimental Platform

- Replay of the Process of Process Modeling



Enables replaying a PPM instance step by step and to reconstruct each intermediate version of the model

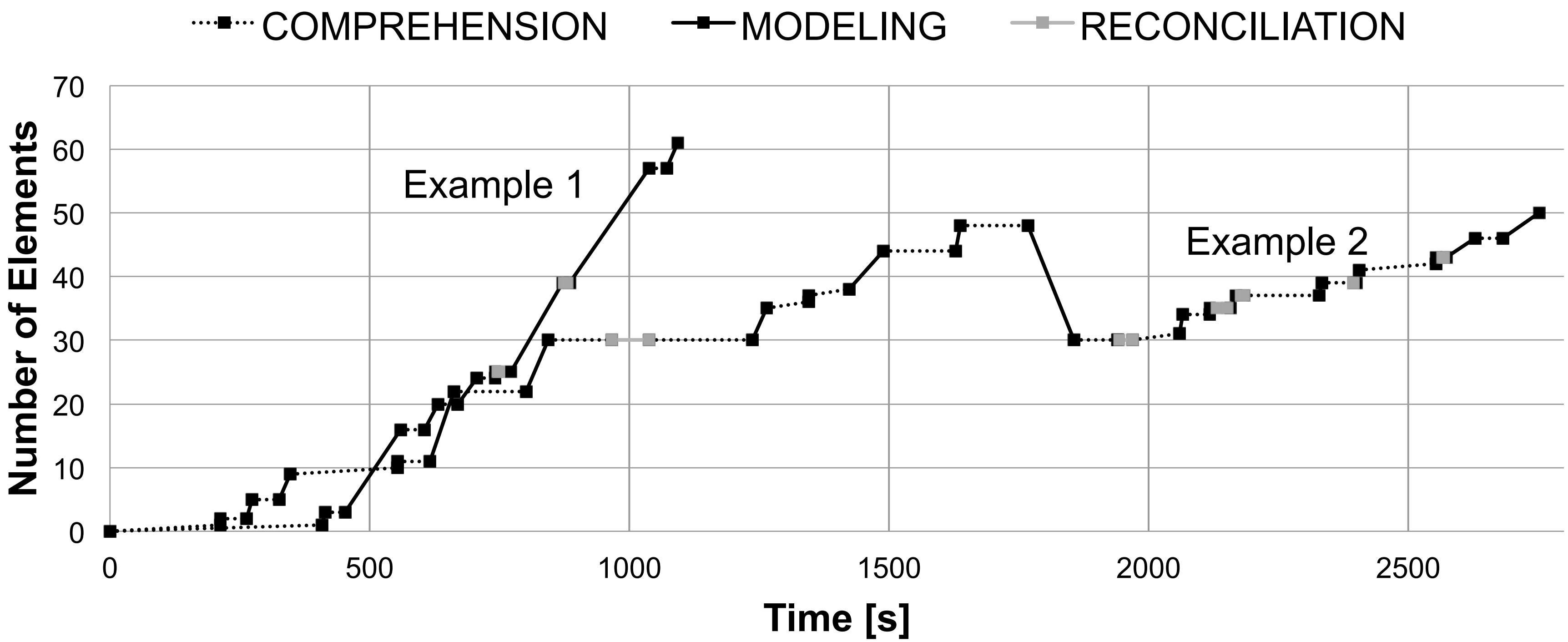
Investigating the Process of Process Modeling
Visualizing the PPM with Dotted Charts



J. Claes, I. Vanderfeesten, J. Pinggera, H. Reijers, B. Weber and G. Poels: A visual analysis of the process of process modeling. Information Systems and e-Business Management 13(1):147–190, 2015.

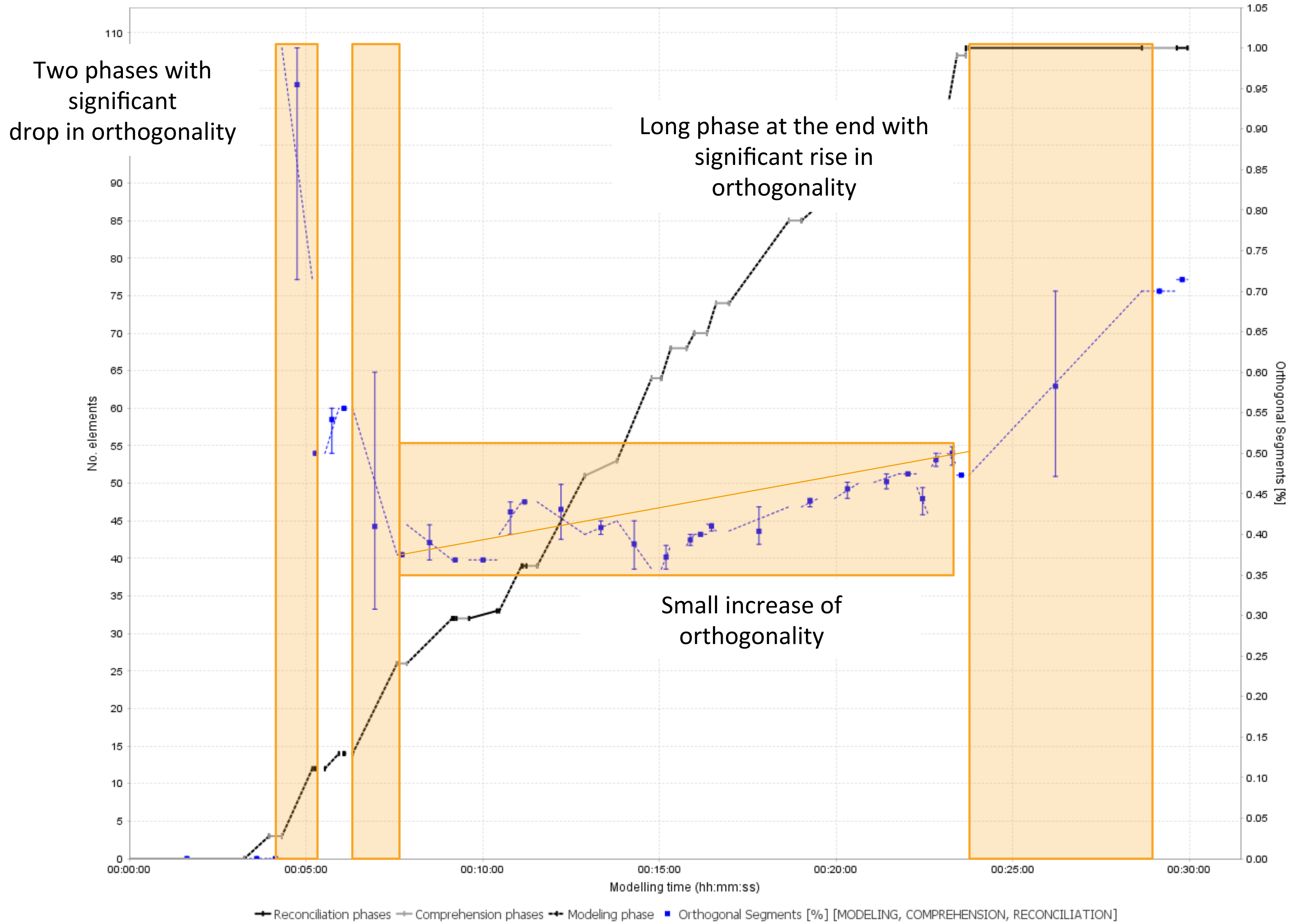
Modeling Phase Diagrams

- Visualize PPM by accumulating model interactions to modeling phases



J. Pinggera, P. Soffer, S. Zugal, B. Weber, M. Weidlich, D. Fahland, H. Reijers and J. Mendling: Modeling Styles in Business Process Modeling. In: Proc. BPMDS ’12, pp. 151–166.

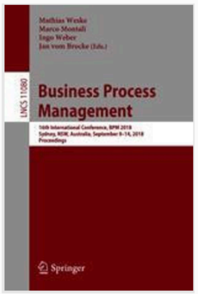
Evolution of the Design Artifact
Example of the Orthogonality Property



Modeling Mind
(FWF P26609-N15)

ModErARe
(FWF P26140-N15)

Classifying Modelers based on Pragmatic Modeling Features



International Conference on Business Process Management
BPM 2018: Business Process Management pp 322-338 | Cite as

Who Is Behind the Model? Classifying Modelers Based on Pragmatic Model Features

Authors | Authors and affiliations

Andrea Burattin, Pnina Soffer, Dirk Fahland, Jan Mendling, Hajo A. Reijers, Irene Vanderfeesten, Matthias Weidlich, Barbara Weber

Creation of process models

- Creating process models is a complex cognitive design activity
- To accomplish that, the modeller has to
  - Construct a mental representation of the problem domain
  - Externalize the mental model into a process model

- Modelling is **not** for free: it imposes a substantial cognitive load
  - Cognitive load is a good predictor of task performance
  - Overload causes a drop in performance

Soffer, P., Kaner, M., Wand, Y.: Towards Understanding the Process of Process Modeling: Theoretical and Empirical Considerations. In: Proc. ER-BPM'11. (2012) 357-369

Claes, J., Vanderfeesten, I., Pinggera, J., Reijers, H.A., Weber, B., Poels, G.: Visualizing the Process of Process Modeling with PPM Charts. In: Proc. TAProViz'12. (2013) 744-755

Wickens, C.D., Hollands, J.G.: Engineering Psychology and Human Performance. 3 edn. Pearson (1999)

Experts and novices

- Experts and novices respond differently to model creation tasks
- Novices
  - Challenged in integrating parts of the problem description
  - Challenged in mapping problem description into knowledge structures
  - Challenged in making abstractions (focus on specific functional details)

- …and experts
  - Tend to develop a holistic understanding
  - Abstract from specific problem characteristics
  - Categorize textual descriptions before developing solutions

Batra, D., Davis, J.G.: Conceptual data modelling in database design: similarities and differences between expert and novice designers. International journal of man machine studies 37(1) (1992) 83-101

Narasimha, B., Leung, F.S.: Assisting novice analysts in developing quality conceptual models with UML. Communications of the ACM 49(7) (2006) 108-112

The role of the modelling tool

- Externalization of the mental model is achieved by interacting with a modelling tool
  - Modeller performs a sequence of interactions which results into [intermediate] models
- Differences between experts and novices suggest that modelling tool should provide different kinds of support and guidance

Can a modelling tool distinguish between experts and novices modellers?

Yes.

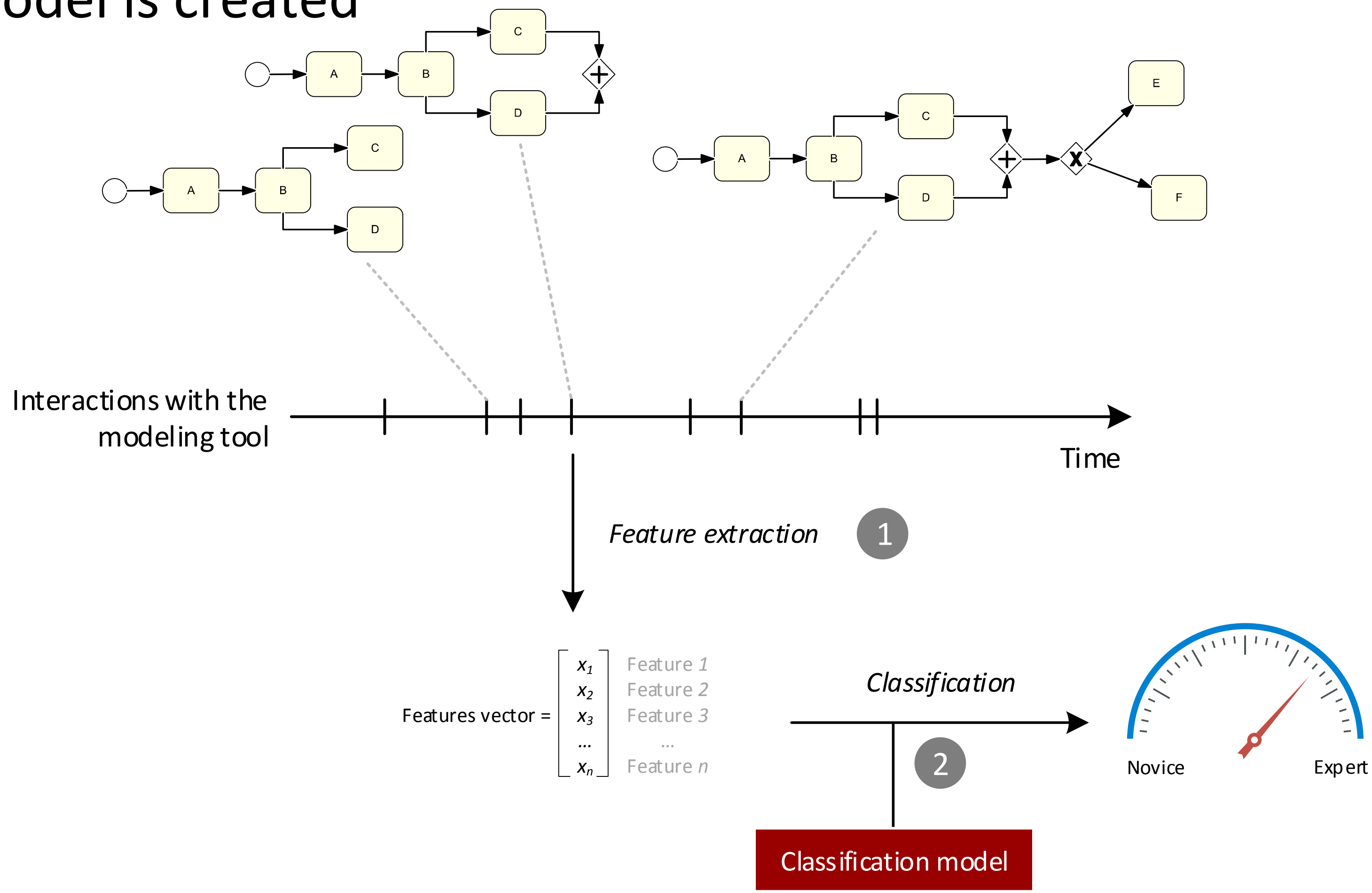
Requirements

- Requirements for expertise prediction
  - R1. Based on objective measures
  - R2. Unobtrusive and no additional effort required
  - R3. Work "online" and applicable to intermediate models
  - R4. Be independent of the modelling tool

- Possible approaches
  - Self-assessment of the user… violates R1 and R3
  - Use a questionnaire to elicit expertise… violates R2 and R3
  - Use neuro-physiological data (e.g., EEG)… violates R2
  - Analyze interactions with modeling platform… violates R4
  - Analyze pragmatic features of (intermediate) artifacts

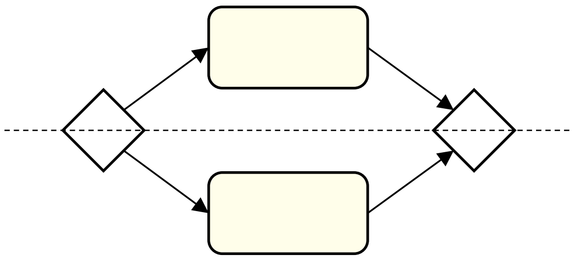
General idea of the approach

- After each interaction with the modelling tool an intermediate model is created

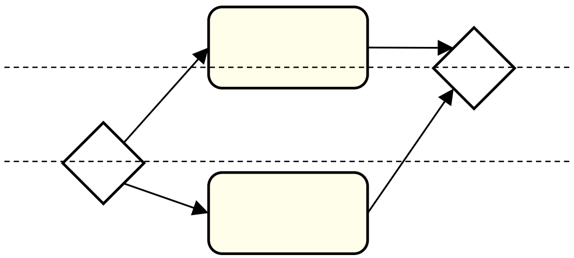


Feature identification

- Given a BPMN model we extract the following pragmatic features
- Features referring to the alignment of elements
  Two nodes are aligned if they share at least one of the coordinates (within a threshold)
  - F1. Percentage of aligned SESE fragments
  - F2. Percentage of activities in aligned SESE fragments
  - F3. Percentage of activities in not-aligned SESE fragments

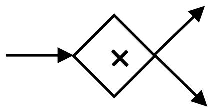


vs.



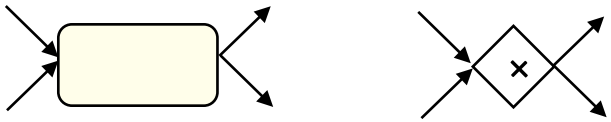
Feature identification (cont.)

- Features referring to the type and usage of gateways
    - F4. Number of explicit gateways



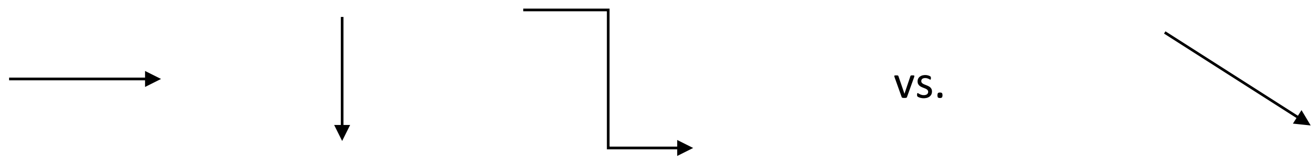
    - F5. Number of implicit gateways

    - F6. Number of reused gateways



Feature identification (cont.)

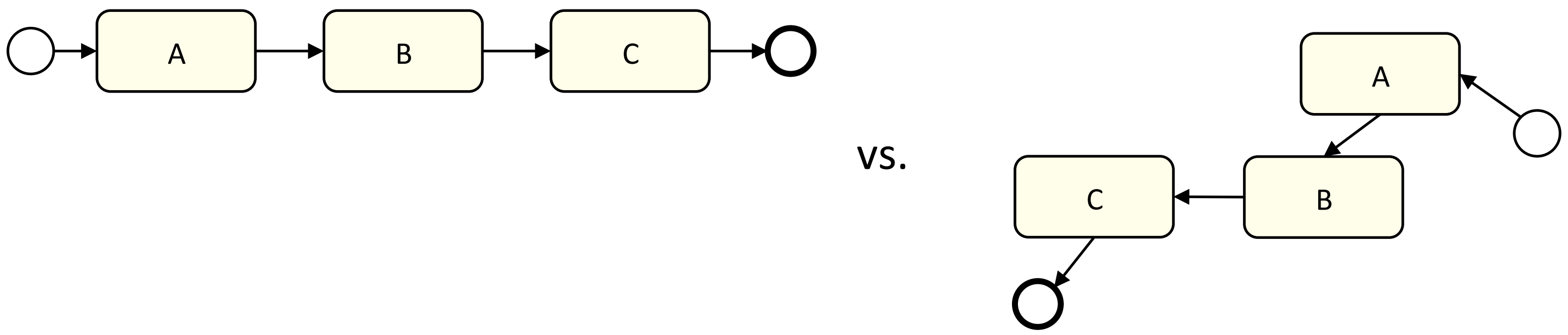
- Features referring to the style of edges
  - F7. Percentage of orthogonal segments



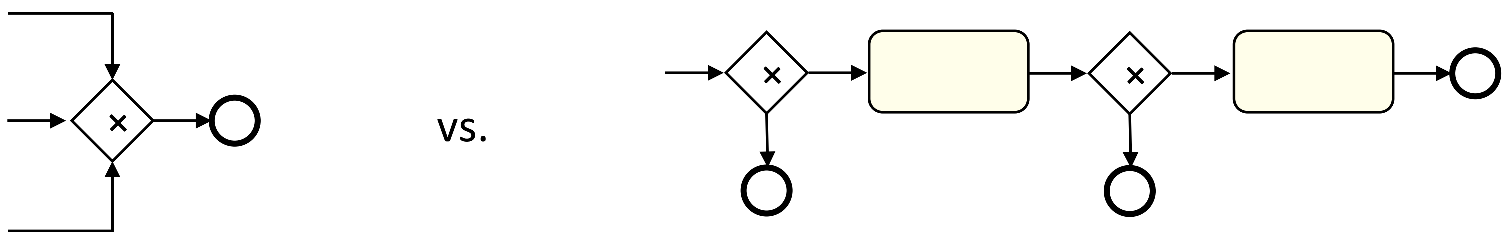
  - F8. Percentage of crossing edges

Feature identification (cont.)

- Feature referring to the process "as a whole"
  - F9. M-BP: consistency of the flow with respect to temporal logical ordering

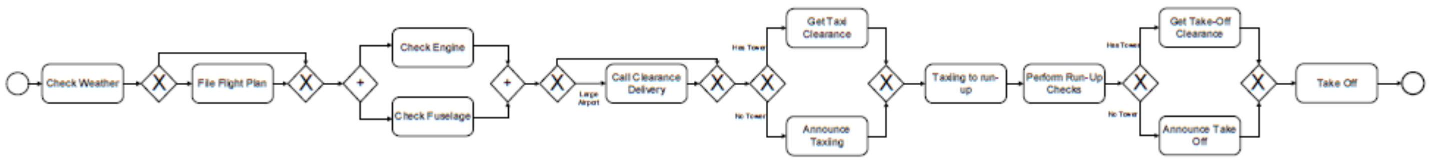


  - F10. Number of ending points

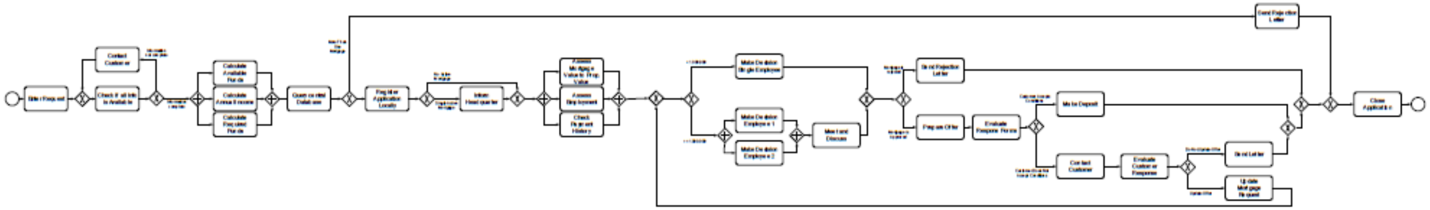


Datasets used for validation

- Two modelling datasets collected during modelling sessions in TU Eindhoven and Berlin in 2010, DOI: 10.5281/zenodo.1194779
- Cheetah as modelling platform
- Subjects were asked to model two processes
  - “pre-flight”, reference:



  - “mortgage-1”, reference:



Datasets used for validation (cont.)

- Number of models and modelling sessions

	Experts		Novices	
	Sessions		**Sessions**	
pre-flight	39		118	
mortgage-1	31		144	

- Mann-Whitney U test (are features significant discriminators of expertise levels?)

		F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
pre-flight	***p***	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001
mortgage-1	***p***	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001

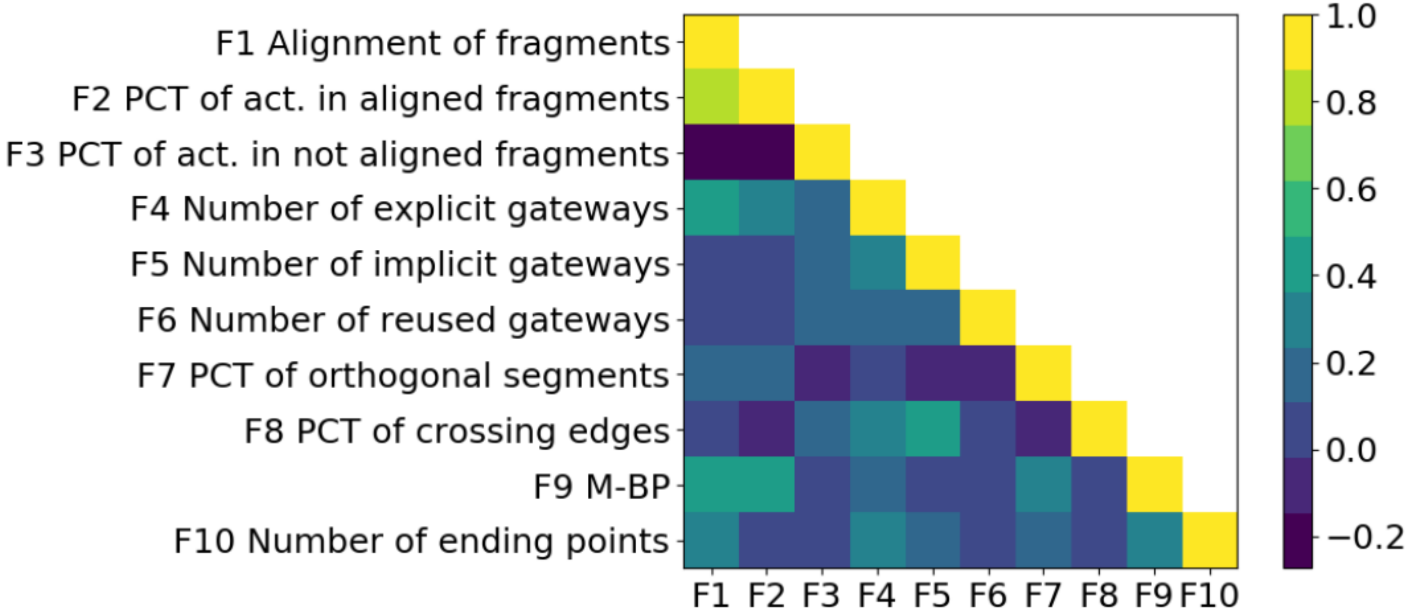
Descriptive statistics (mean)

DIRECTION OF MEAN VALUES SHARED FOR MOST FEATURES

Mean values	**mortgage-1**		**pre-flight**	
	Experts	**Novices**	**Experts**	**Novices**
F1. Alignment of fragments	0.86	0.81	0.82	0.76
F2. % acts in aligned frags	0.46	0.43	0.50	0.44
F3. % acts in not-align frags	0.09	0.10	0.08	0.10
F4. # explicit gateways	11.90	10.19	6.84	5.94
F5. # implicit gateways	1.31	1.58	0.37	0.49
F6. # reused gateways	0.34	0.32	0.50	0.47
F7. % orthogonal segments	0.71	0.60	0.57	0.49
F8. % crossing edges	0.01	0.02	0.012	0.008
F9. Flow consistency	0.95	0.88	0.95	0.91
F10. # end points	2.74	2.27	1.60	1.64

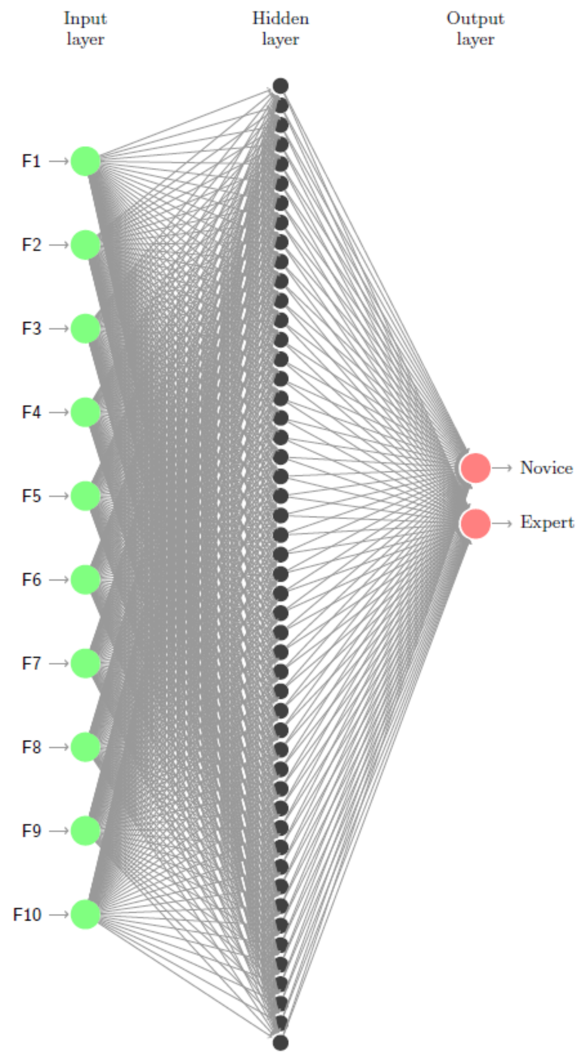
Descriptive statistics (correlations)

- Pearson correlation coefficient of features
  - Little indication of correlation: features capture complementary aspects



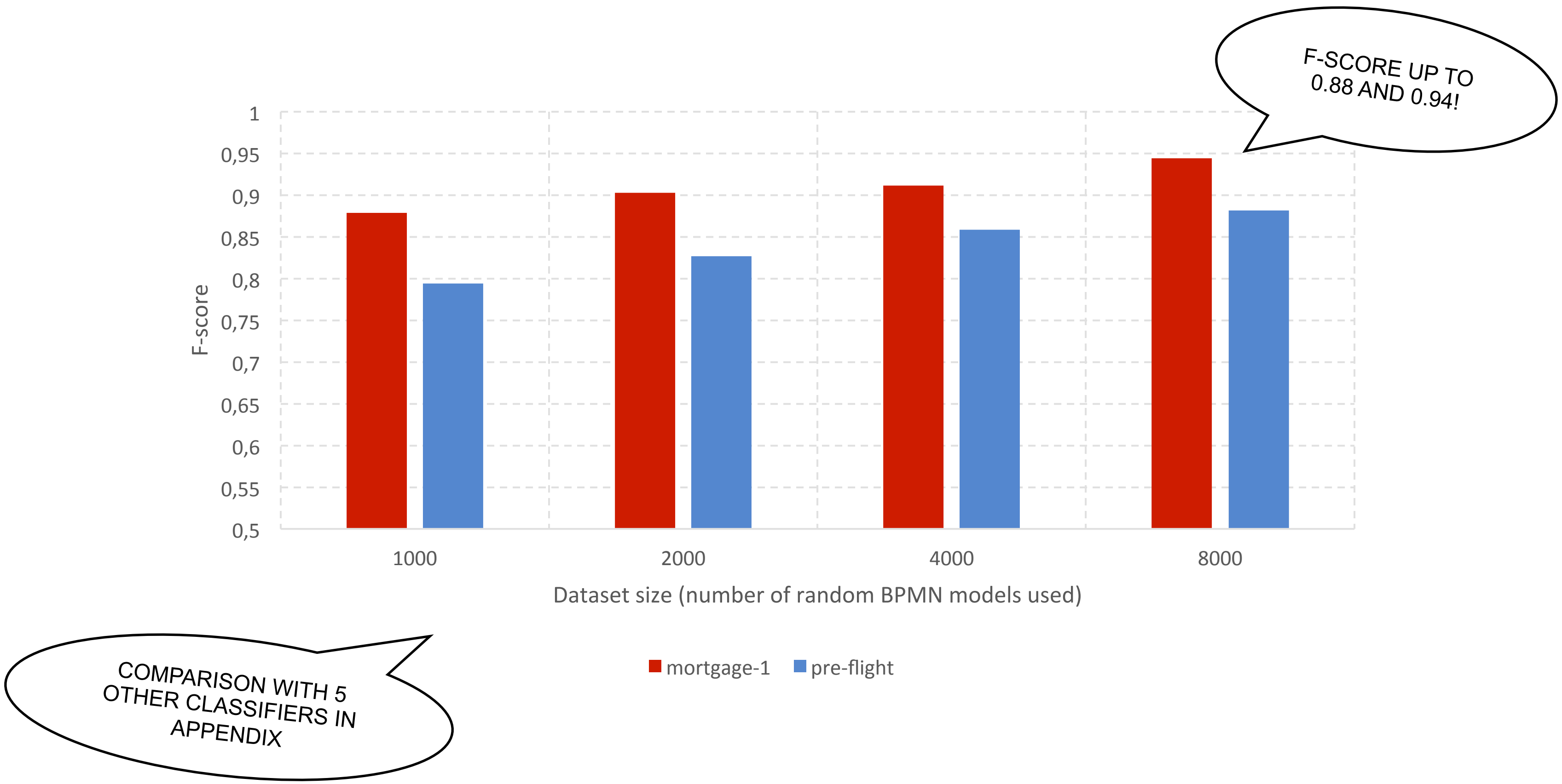
Problem as classification

- Classification problem
  - Input: 10-dimensional feature space (one for each feature)
    - Each intermediate model as independent model sample
    - Only models from the last 70% of the modelling session (to avoid almost-empty models)
  - Output: likelihood of classification of each class
- We used a feed forward neural network with a hidden layer with 50 neurons
  - Training with Multilayer Perceptron
  - Software based on Weka, available at github.com/DTU-SPE/ExpertisePredictor4BPMN



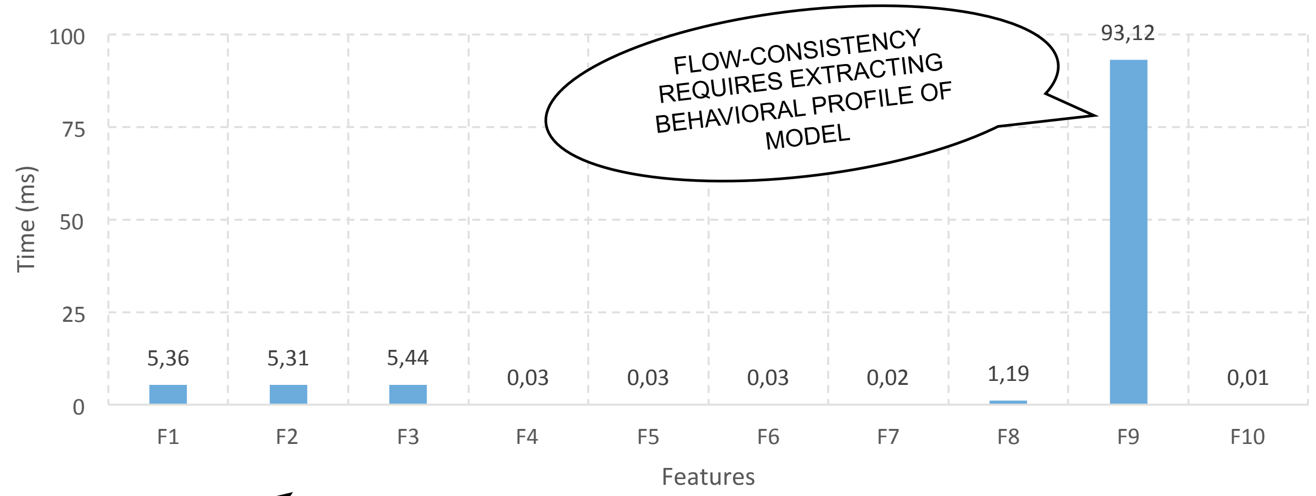
Classification performance

- Tests on random datasets of (intermediate) BPMN model
  - Quality in terms of F1: harmonic mean between *precision* and *recall*
  - Results as average of each of the 10-fold cross validation



Time performance

- Time required to compute each of the 10 features
  - Standard Java implementation (Cheetah) on standard laptop
  - Tests with typical PC usage maintained (to simulate modeller settings)
- Average time over 18k samples from biggest dataset (mortgage-1)



ALL FEATURES IN ABOUT 110 MS: FASTER THAN MODEL UPDATES!

Conclusion and limitations

- We presented an approach to classify modellers
  - Decision is based on objective measures
  - Decision according to artifacts being modelled
  - Fast computation, applicable to intermediate models
- Identified requirements are all met
- Classification results as F-score
  - On mortgage-1: 0.94
  - On pre-flight: 0.88 (the process lacks complex behavioural structures)
- Limitations
  - Currently only applicable to BPMN models
  - Big models (> 30 activities) might require more time to compute features
  - Same modelling task used for training and prediction

R1. Based on objective measures
R2. Unobtrusive and no additional effort
R3. "Online" and intermediate models
R4. Independent of the modelling tool

Impact and future work

- Potential impact on several aspects
  - For developers: design tools that adapt themselves to the user
  - For educators: assess user capabilities and form groups
  - For practitioners: recruitment, task allocation and team formation

- Future work include
  - Generalizing the task to predict models not used for training
  - Improve prediction of sessions rather than models
  - Continue the feature engineering process

Towards a Neuro-adaptive Modeling Platform

DTU Compute
Department of Applied Mathematics and Computer Science

Overview of a Neuro-adaptive System

2.) Biological signal is analyzed to derive a mental state

3.) System adapts on the base of the user's mental state and context

1.) Biological Signal (e.g., EEG, pupil dilation, HRV, skin conductance) is recorded from the user

Riedl, R., Léger, P.-M.: Fundamentals of NeuroIS. Springer Berlin Heidelberg, Berlin, Heidelberg (2016).

Cognitive Load During Development Activities

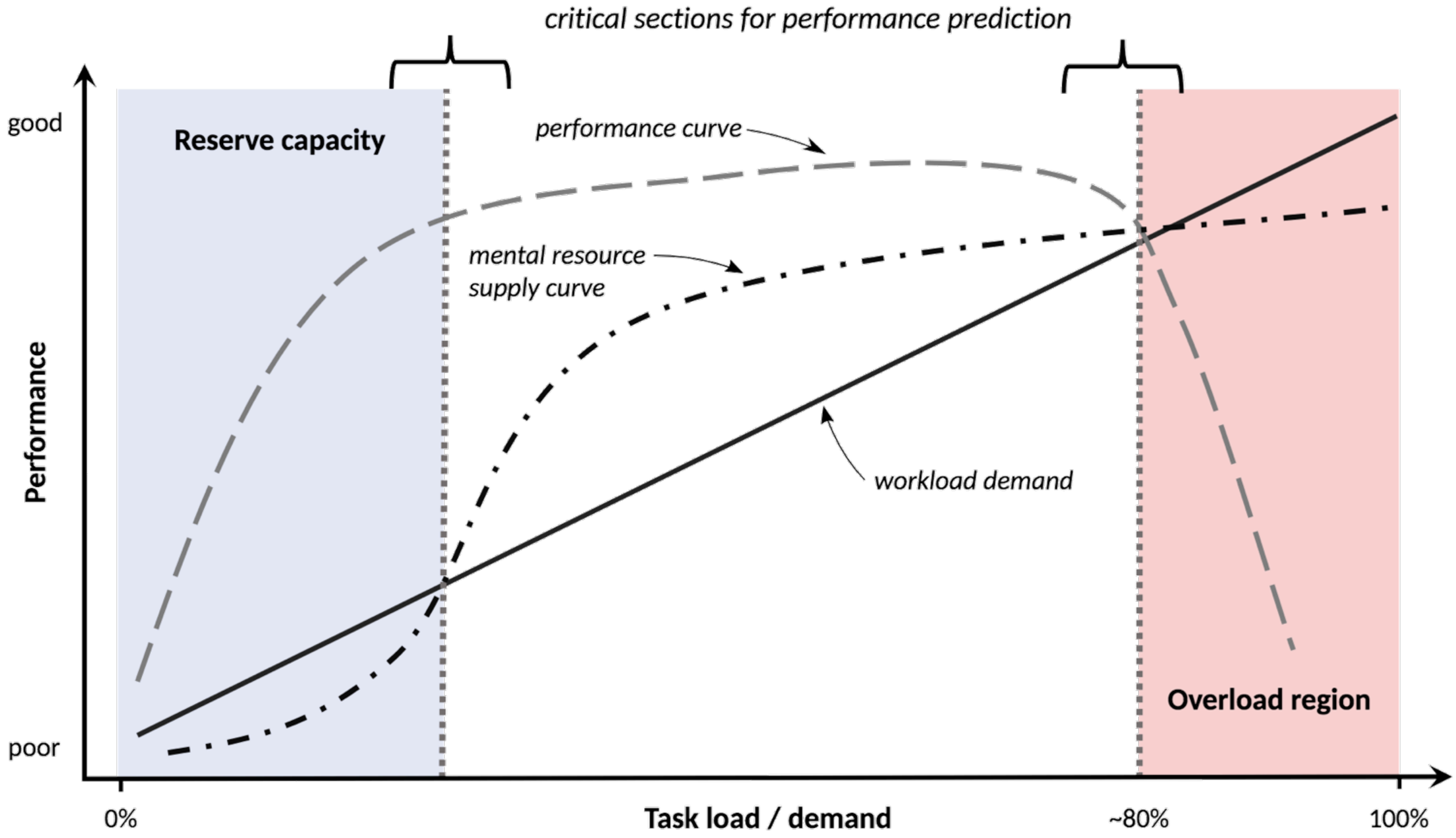
- Cognitive load (CL) characterizes the demands tasks impose on the limited information processing capacity of the brain

- High CL leads to poor task performance and to wrong decisions

Chen, F., Zhou, J., Wang, Y., Yu, K., Arshad, S. Z., Khawaji, A., & Conway,D.: Robust Multimodal Cognitive Load Measurement. Springer 2016.

Wickens, C. D., & Hollands, J. G.: Engineering Psychology and Human Performance (3rd ed.). Pearson 1999.

Cognitive Load: A Predictor for Task Performance



Chen, F., Zhou, J., Wang, Y., Yu, K., Arshad, S. Z., Khawaji, A., & Conway,D.: Robust Multimodal Cognitive Load Measurement. Springer 2016.

Assessment of Cognitive Load

- **Subjective ratings**
  - SWAT, NASA-TLX
- **Performance measures**
  - Dual-task setting
- **Behavioral measures**
  - Eye tracking, i.e., eye movements
  - User interactions
- **Neuro-physiological measures**
  - Heart rate variability
  - Eye tracking, i.e., pupillary responses, eye blink rate
  - EEG
  - Galvanic Skin Response

Chen, F., Zhou, J., Wang, Y., Yu, K., Arshad, S. Z., Khawaji, A., & Conway,D.: Robust Multimodal Cognitive Load Measurement. Springer 2016.

Overview of a Neuro-adaptive System

2.) Biological signal is analyzed to derive a mental state

3.) System adapts on the base of the user's mental state and context

1.) Biological Signal (e.g., EEG, pupil dilation, HRV, skin conductance) is recorded from the user

Riedl, R., Léger, P.-M.: Fundamentals of NeuroIS. Springer Berlin Heidelberg, Berlin, Heidelberg (2016).

Overview of a Neuro-adaptive System

2.) Biological signal is analyzed to derive a mental state

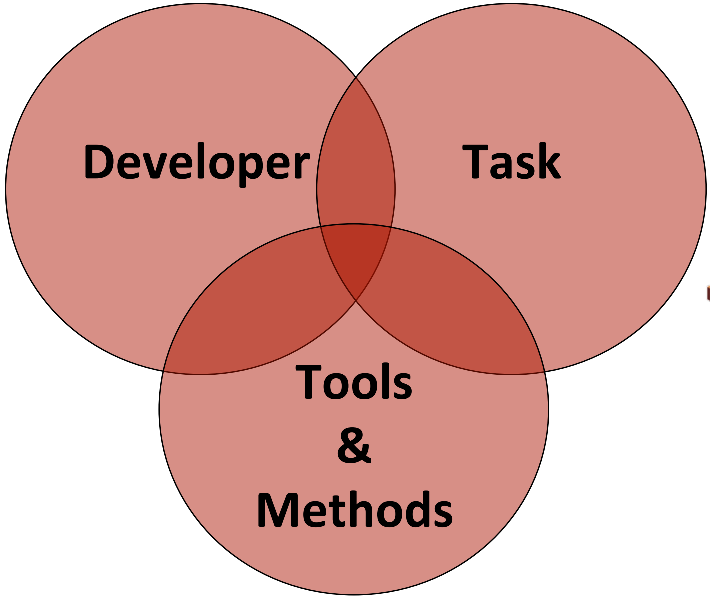
3.) System adapts on the base of the user's mental state and context

1.) Biological Signal (e.g., EEG, pupil dilation, HRV, skin conductance) is recorded from the user

Riedl, R., Léger, P.-M.: Fundamentals of NeuroIS. Springer Berlin Heidelberg, Berlin, Heidelberg (2016).

Relevant Context Factors for Development Activities

- Factors influencing Cognitive Load and Task Performance



Digital Artifact

Developer-specific factors including expertise, domain knowledge, cognitive abilities

Task-specific factors including inherent task complexity, task representation

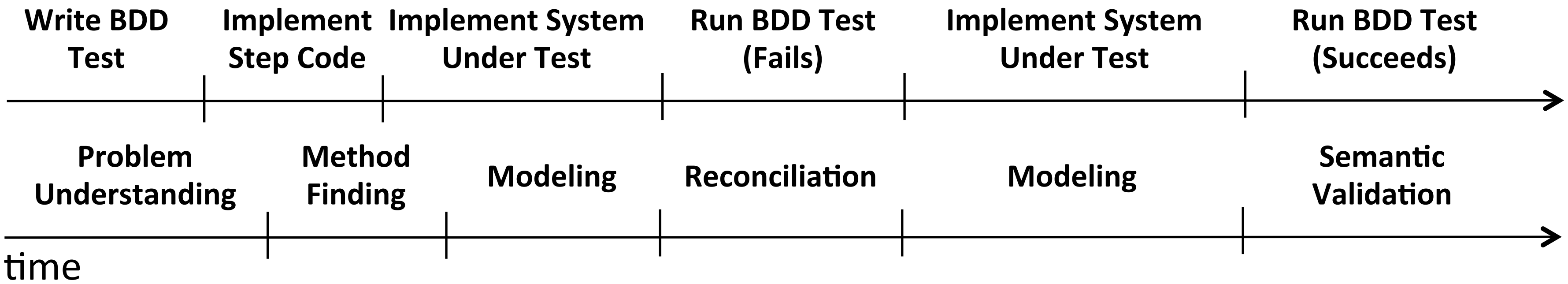
Tool & Method-specific factors including development platform (i.e., language, tool support)

Not sufficient!

Relevant Context Factors for Development Activities

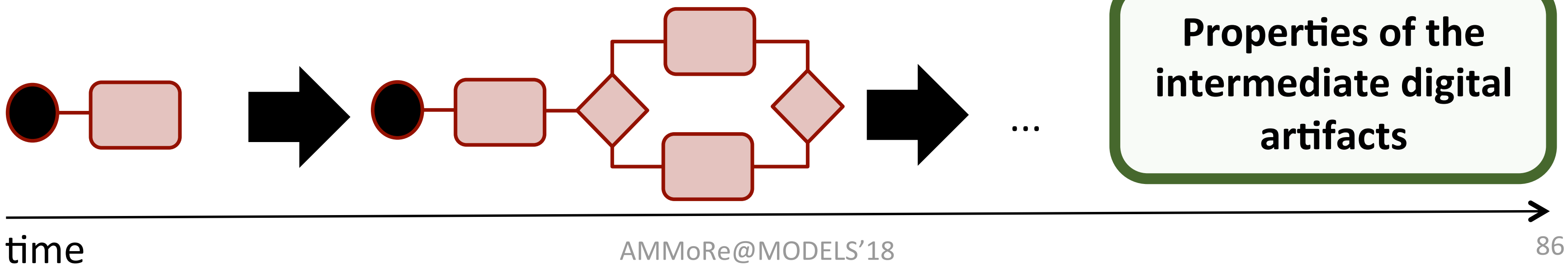
- Development activities are characterized by flexible processes
  - Repeated execution of different phases

Process of creating the digital artifact

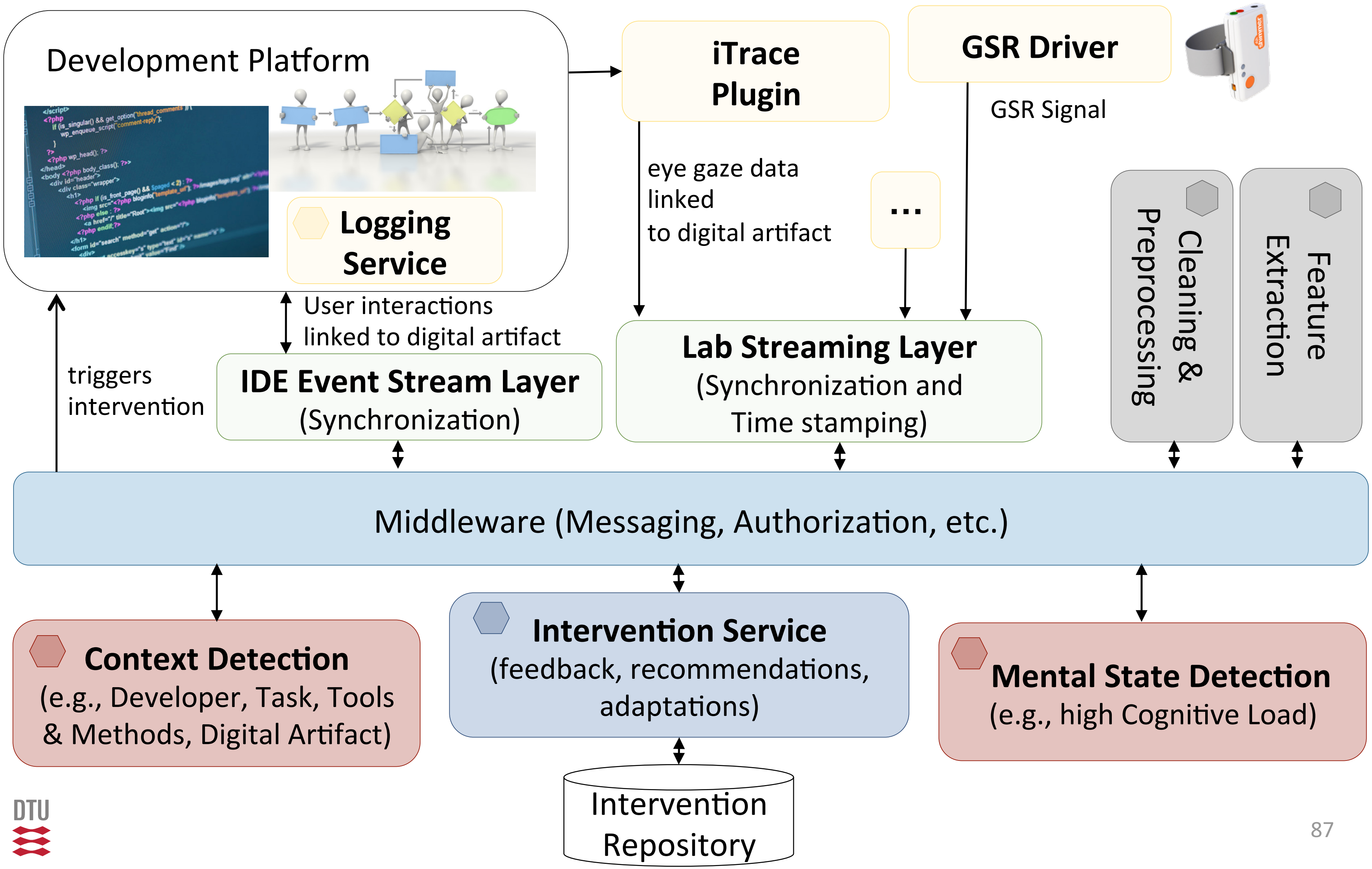


- The digital artifact evolves from an initial state over a set of intermediate versions to the final end product

Properties of the intermediate digital artifacts

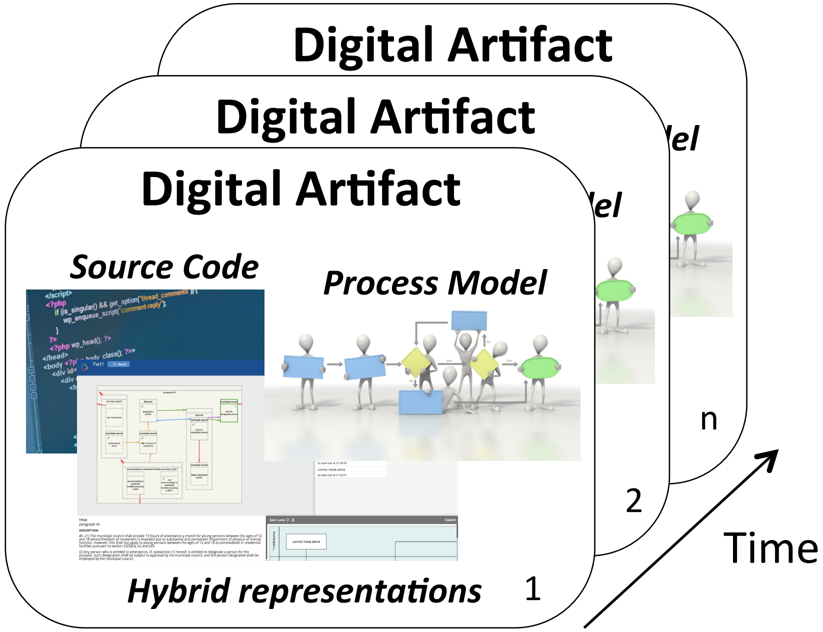


Neuro-adaptive Development Platform: Suggested Software Architecture

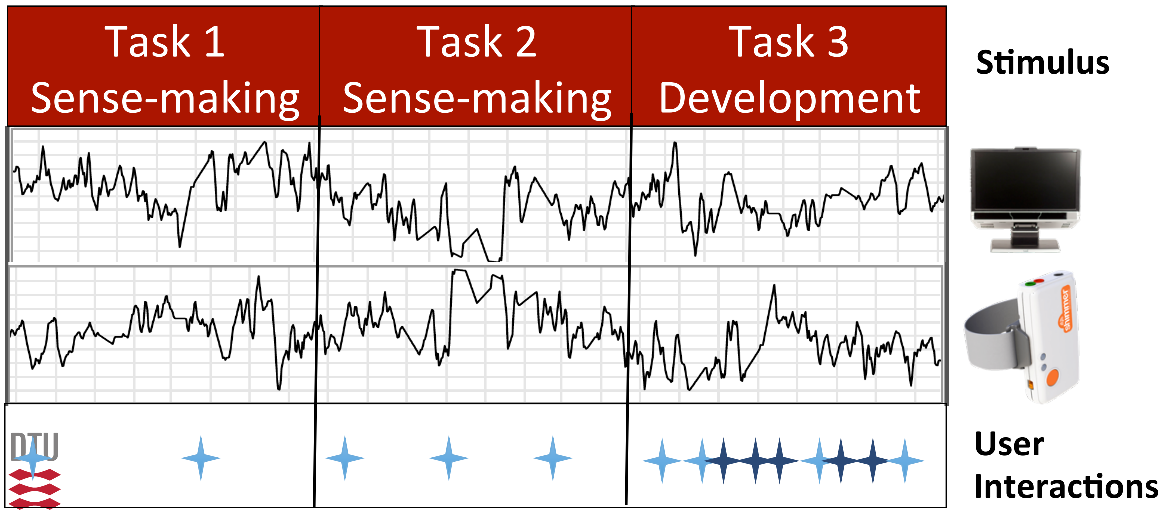


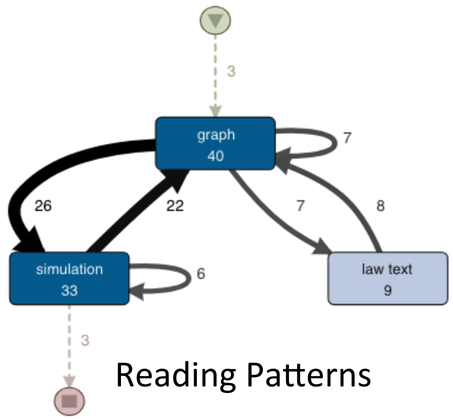
DTU

New Avenues for Investigating the Evolution of a Digital Artifact



- **Process-oriented** through continuous data collection
- **Multi-modal** data collection
- Measurements **linked with digital artifact**





Summary

- Enabling **flexibility** in **executable process models** through process adaptation, process evolution, and business process variability

- Investigating the **process** of process modeling
  - Cheetah Experimental Platform
    - Logging User Interactions
    - Properties of the Design Artifact

- From offline to **on-the fly**
  - Example: On the fly classification of modelers

- Going **beyond the artifact** and towards **multi modal measurements**
  - Example: Neuro-adaptive modeling support

Thanks for your attention!

bweb@dtu.dk