UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT

BUSINESS RULES & BUSINESS RULES MANAGEMENT
PART 2: DMN

AGENDA

- Recap
- Business Rules Management
  - Organizing
    - .. In DMN
    - .. In Declare
  - Implementing

RECAP

- Capturing: Elicitation of rules
  - Identification, rule mining
- Authoring: Formalization of business rules
  - Model-Level (Declare)
  - Behavior of Instances (DMN)

- Plan
  - Identify company processes/legal domains
  - Assign human ressources
- Capture
  - Identify: What are the relevant business rules?
- Author
  - Formalize Business Rules
- Organize
  - Ensure the formalized set of business rules is correct/sound
- Apply
  - Utilize business rules to ensure compliant processes/models

AGENDA

- Recap
- Business Rules Management
  - Organizing
    - .. In DMN
    - .. In Declare
  - Implementing

BUSINESS RULES MANAGEMENT

- So far, rules were captured and authored
- Rule bases are often authored incrementally, by many modelers
- Distributed business rules management may cause errors in business rules, e.g. redundancies or inconsistencies
- We need to avoid such inconsistencies, otherwise processes may produce erroneous results

REMINDER

- Reminder – Business Rules…
  - concerning the structure and common behavior (of models)
    - Formalism I: Declare
  - concerning the behavior of single instances
    - Formalism II: Decision Model and Notation (DMN)

EXAMPLE ERROR IN AUTHORING

REGARDING THE MODEL ITSELF

- Modeller 1: Response(A,B)
- Modeller 2: NotResponse(A,B)

- Problem: If we have a task A, no possible process model can satisfy the set of business rules

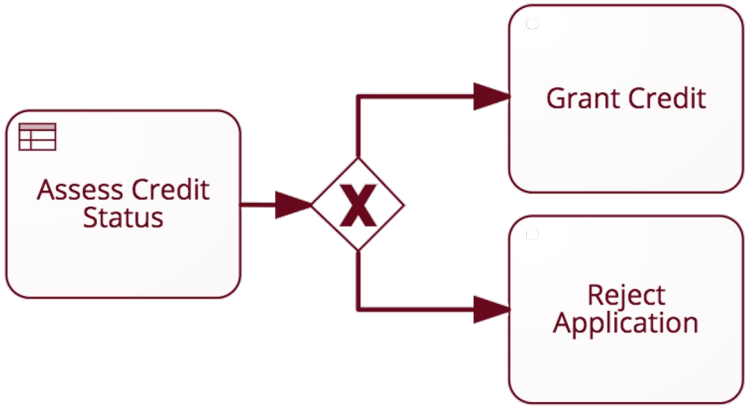
EXAMPLE ERROR IN AUTHORING

REGARDING THE BEHAVIOR

- Modeler A and B enter two rules in a DMN table:

Input	Output
Name	**CreditWorthy?**
Bob	true
Bob	false

- What is the outcome?
- No inferences possible
- Rule base cannot be used as intended



ERRORS IN BUSINESS RULES

- Central assumption: Mistakes happen, due to nature of collaborative work
- We will very likely find inconsistencies in the rule bases of companies
- What can we do with these inconsistencies?

- Simply delete inconsistent rules?
- No, because simply deleting business rules might cause problems even more severe than inconsistencies (cf. consequences of non-compliant processes!)

BUSINESS RULES MANAGEMENT

SEVERAL THINGS TO DO IN ORGANIZING

- Find the rules that are inconsistent between each other (sometimes across several rules!)
- Asses the severity of the inconsistency caused by each inconsistent rule
- Visualize the inconsistencies so analysts get a decision support how to resolve the inconsistencies

AGENDA

- Recap
- Business Rules Management
  - Organizing
    - .. In DMN
    - .. In Declare
  - Implementing

SOURCES OF ERRORS IN DMN

- Collaborative Modelling
  - Different views on the same domain of interest
- Redundant Information
- Incomplete Information
- Inconsistent Information

REPORTS FROM THE FIELD

- Batoulis, K., Nesterenko, A., Repitsch, G., & Weske, M. (2017). Decision Management in the Insurance Industry: Standards and Tools. In *BPM (Industry Track)* (pp. 52-63).
- Analyzed DMN rule set of large insurance industry
- Found that 30% of analyzed rules were erroneous!

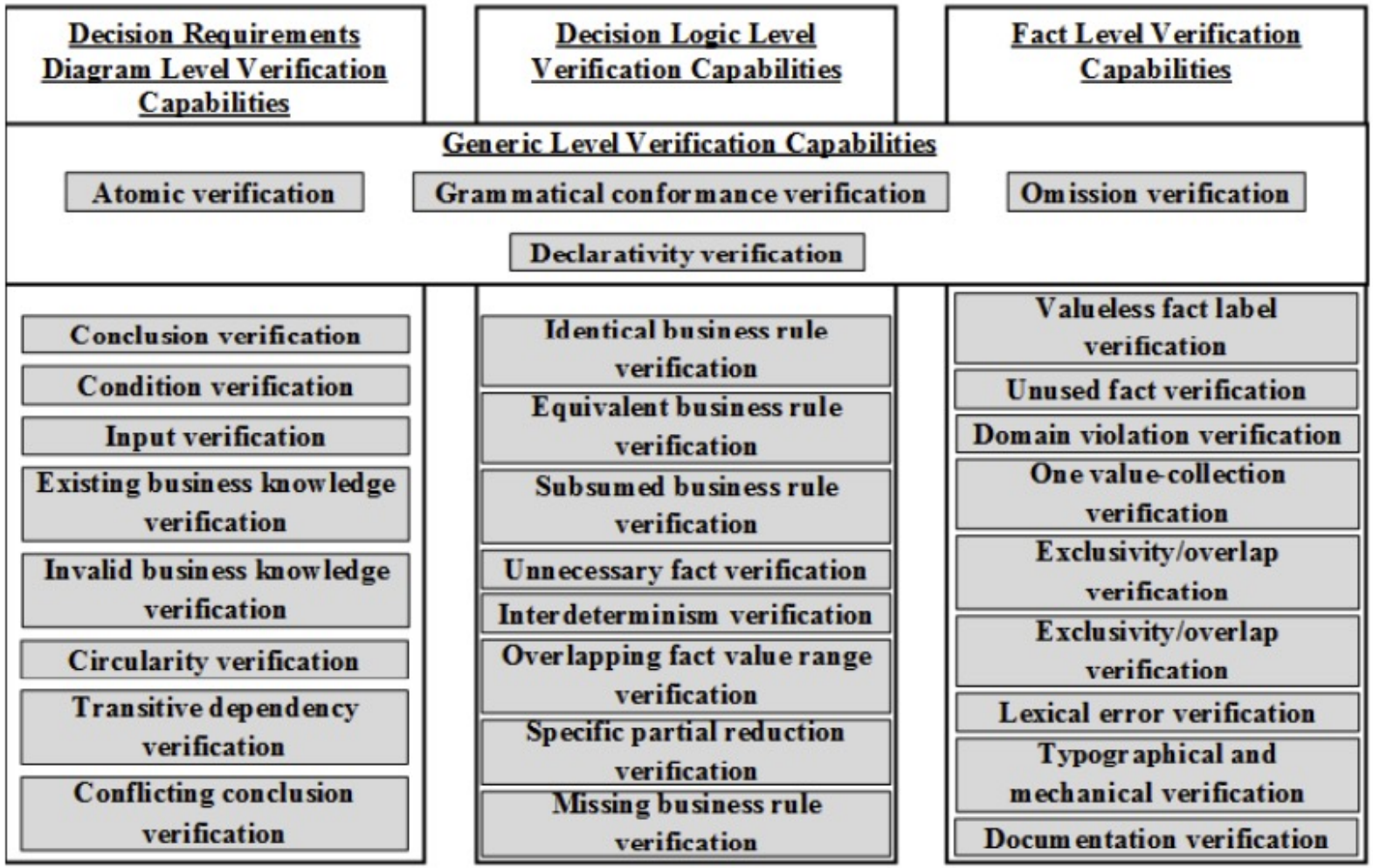
ERROR TYPES IN DMN

- We know we can expext „errors“ in DMN tables. But how exactly do they look?
- Your turn: *What can be error types in DMN?*
  - *Redundancy*
  - *Inconsistency*
  - *Overlapping rules*
  - …..
- Problem: We need to find „all“ errors in organizing. Yet, how do we know what to look for?

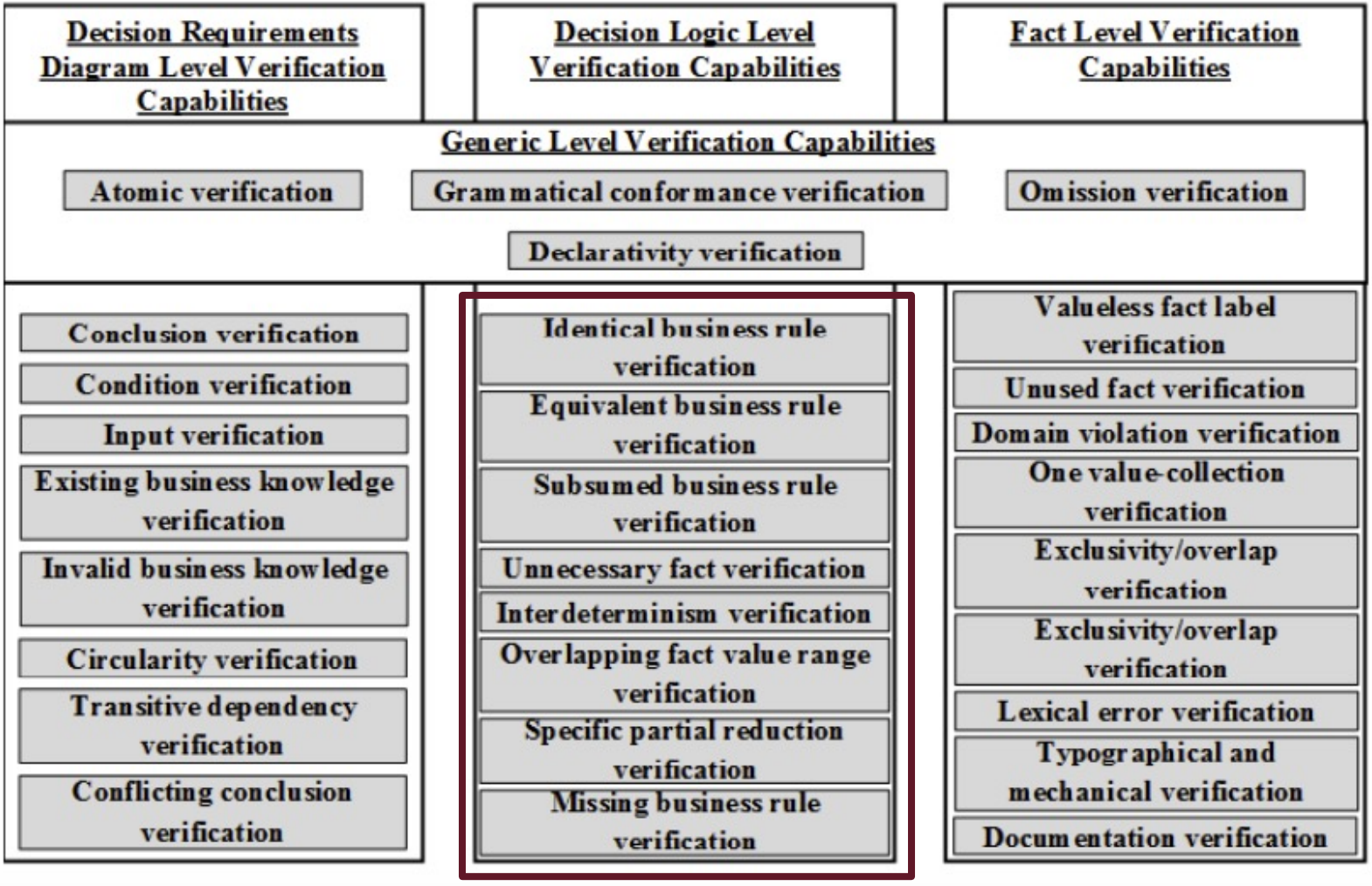
ERROR TYPES IN DMN

- Example: Qualitative Research based on expert interviews (Smit et al. (2018))
- Idea: Conduct expert panels with multiple companies to gain insights on problems encountered in practice
  - Analyse and Synthesis the interview data to distill a classification of error types
- Result: DMN Verification Capability Framework

DMN VERIFICATION CAPABILITY FRAMEWORK



DMN VERIFICATION CAPABILITY FRAMEWORK



DECISION LOGIC LEVEL VERIFICATION

- **Identical rule verification.** Detecting rules which have an identical input, i.e. are redundant.
- **Equivalent rule verification.** Detecting rules which are not identical, but still semantically equivalent. Here, our tool can verify if there exist multiple rules which use synonyms as inputs and are therefore equivalent, based on synonym relations via Wordnet.
- **Subsumed rule verification.** Detecting individual rules which are subsumed by other rules, i.e. they are not necessary. For example, rules containing wildcards often render more specific rules unnecessary due to subsumption.
- **Interdeterminism verification.** Detecting rules which will always be activated together, but have differing or contradicting conclusions. For example, rules which will be activated together must not yield that a customer is both credit worthy, and not creditworthy, as this is logically inconsistent.

DECISION LOGIC LEVEL VERIFICATION

- **Partial reduction verification.** Checking wether ranges can be combined to simplify decision tables.
- **Overlapping condition verification.** Detecting whether there are any overlaps in rule conditions.
- **Unnecessary Facts Verification.** Detecting facts (case-data) that are not used in the decision-making
- **Missing rule verification.** Detecting whether there are any missing business rules, i.e. if there are rules missing for expected inputs.

DECISION LOGIC LEVEL VERIFICATION

IDENTICAL

Identical			
Decision_13nychf			
U	Input +		Output +
	testNumber	testString	-
	integer	string	string
1	=10	"Hello"	-
2	=10	"Nicole"	-
3	=10	"Hello"	-
4	=11	-	-
5	=11	-	-
+	-	-	-

DECISION LOGIC LEVEL VERIFICATION

EQUIVALENT

Decision_1be8xb9

U	Input +	Output +
	-	-
	string	string
1	"invoice"	-
2	"bill"	-
3	"invoice"	-
4	"exam"	-
5	"examination"	-
6	"organisation"	-
7	"organization"	-

DECISION LOGIC LEVEL VERIFICATION

SUBSUMED

Subsumption 1

`Decision_1vo386g`

U	Input +	Output +
	-	-
	integer	string
1	[0..10]	"Hello"
2	[0..5]	"World"
3	[7..9]	"Hello"
4	[7..15]	"FuBa"

DECISION LOGIC LEVEL VERIFICATION

REDUCTION

Decision_0m4xor4

U	Input +		Output +
	–	–	–
	integer	integer	string
1	<0	-	"a"
2	>=0	-	"a"
3	[10..20[	-	"b"
4	[20..30]	-	"b"

DECISION LOGIC LEVEL VERIFICATION

OVERLAPPING

Decision_1iijf13

U	Input +		Output +
	–	–	–
	integer	string	string
1	[0..10]	"a"	-
2	[5..20]	"a"	-
3	[15..30]	-	-

DECISION LOGIC LEVEL VERIFICATION

MISSING

Decision_13hi6aw

U	Input +		Output +
	x	y	–
	integer	integer	string
1	[2..4]	[3..5]	-
2	[6..7]	[2..7]	-

DECISION LOGIC LEVEL VERIFICATION

DEMO

AGENDA

- Recap
- Business Rules Management
  - Organizing
    - .. In DMN
    - .. In Declare
  - Implementing

INCONSISTENCY IN DECLARE

- Consider the following exemplary rule sets

Example 1

RESPONSE(A, B)
NOT RESPONSE(A, B)

Example 2

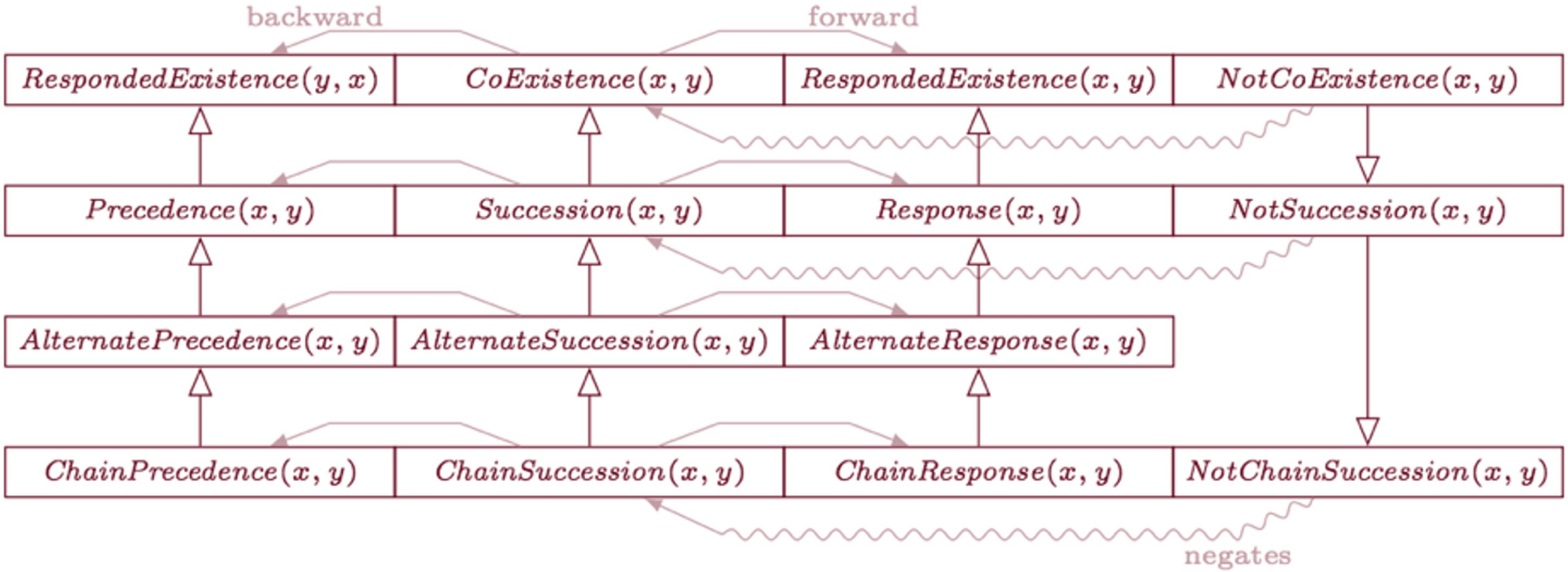
CHAIN RESPONSE(A, B)
NOT RESPONSE(A, B)

Example 3

RESPONSE(A, B)
RESPONSE(B, C)
NOT RESPONSE(A, C)

INCONSISTENCY IN DECLARE

- Template Relations



INCONSISTENCY IN DECLARE

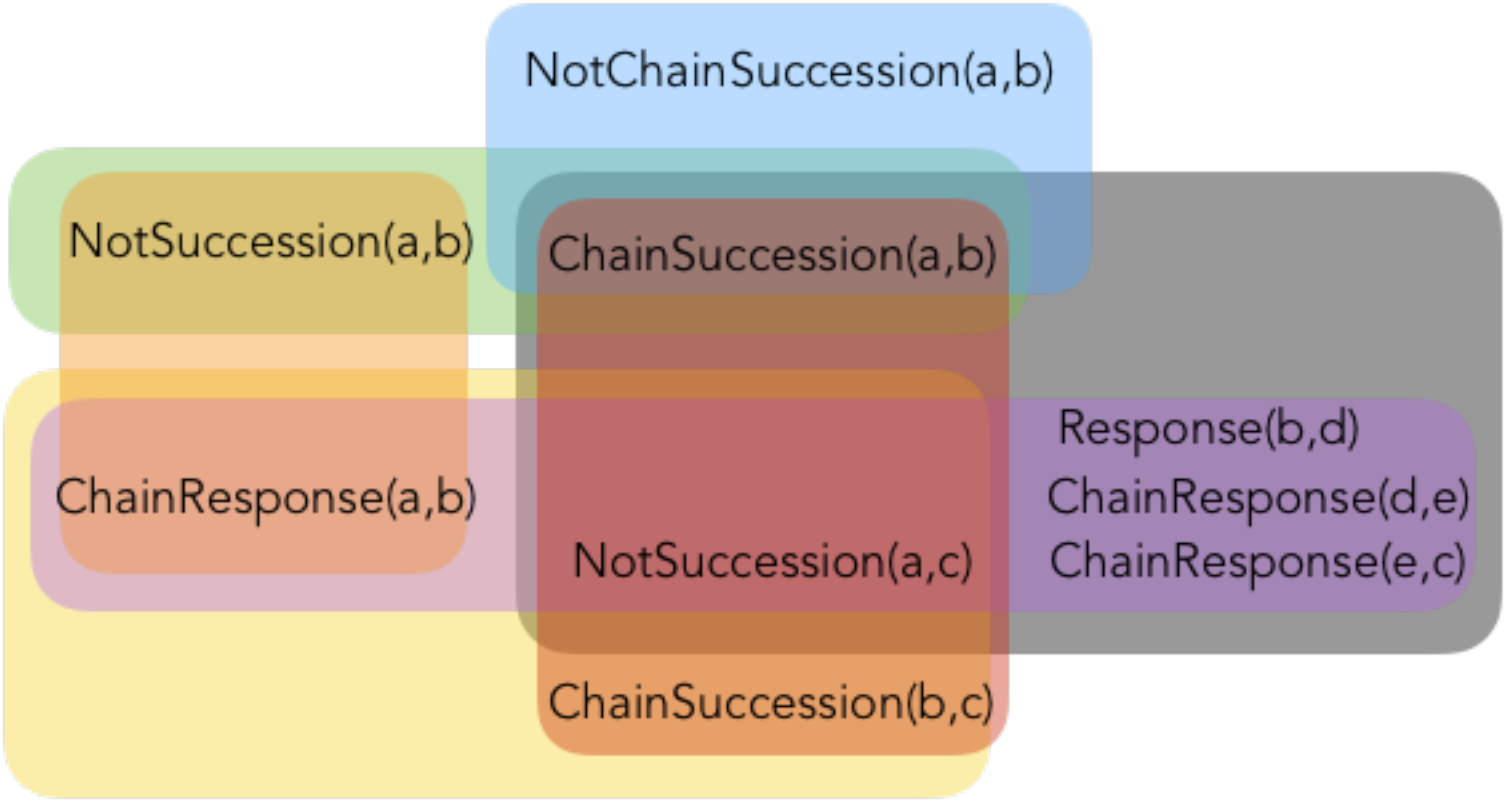
- Goal:
  - Identify Inconsistencies
  - Quantify Inconsistencies (As Guidance for experts)

INCONSISTENCY IN DECLARE

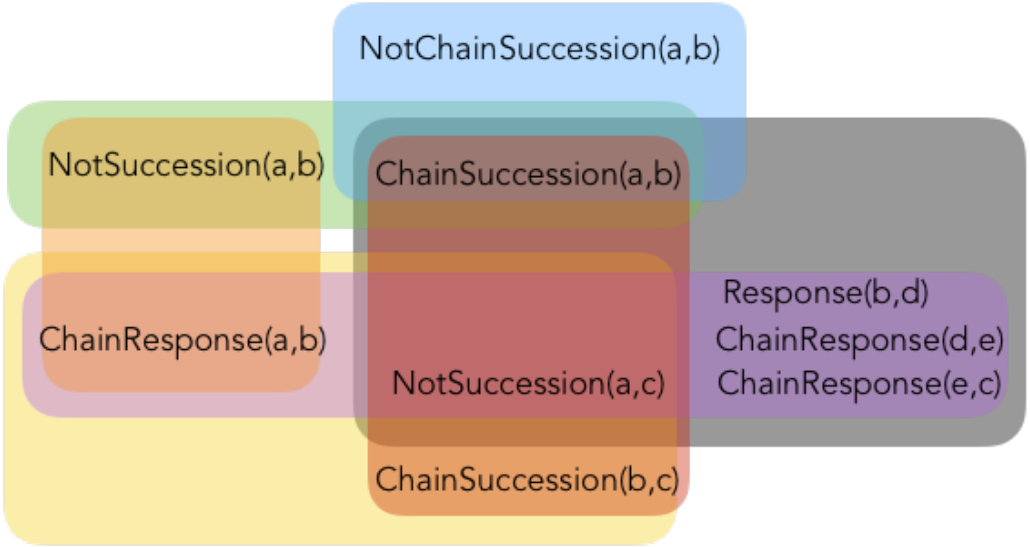
- Minimal Contradictory Subets. Given a declarative process model $M = (A, T, C)$, the set of minimal contradictory subsets MCS(M) is defined via

  $MCS(M) = \{C' \subseteq C \mid C' \text{ is contradictory and minimal in terms of set inclusion}\}$

INCONSISTENCY IN DECLARE



INCONSISTENCY MEASUREMENT



- How (to which degree) inconsistent is the entire rule base?
- How (to which degree) responsible are individual rules, in the context of the overall inconsistent rule base?

INCONSISTENCY MEASUREMENT

THE ENTIRE RULE BASE

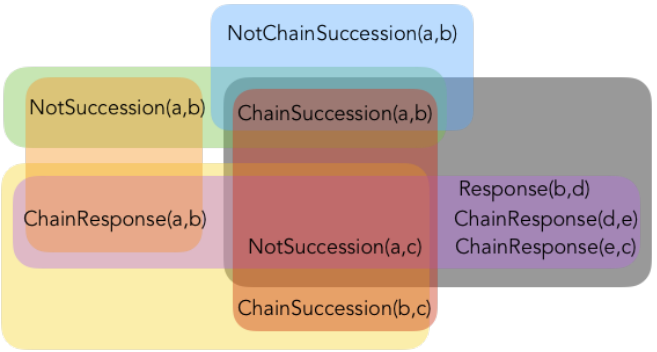
- Inconsistency Measures
- Let $\mathcal{B}$ be a set of all possible business rule bases, then an inconsistency measure I is defined as

  $I: \mathcal{B} \longmapsto [0,\infty[$

- Example: so-called MI-inconsistency measure I_{MI}, defined via

  $I_{MI}(M) = |MCS(M)|$

- i.e., $I_{MI}(Example) = 7$



REPORTS FROM THE FIELD

- Carl Corea, Patrick Delfmann. *Quasi-Inconsistency in Declarative Process Models*. In Business Process Management Forum co-located with the 17th International Conference on Business Process Management (BPM 2019). Wien, 2019.
- Mined Declare rule set from real-life event logs

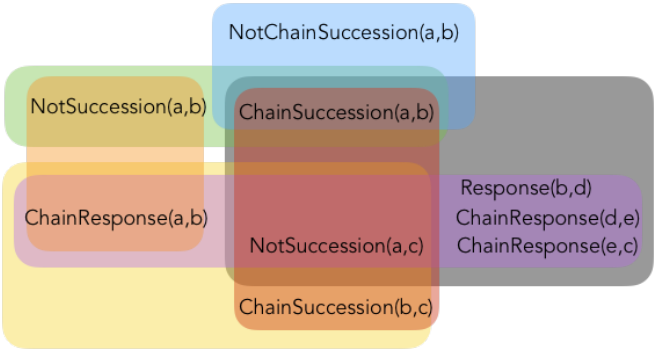
Log	BPI Challenge '17	BPI Challenge '18	Sepsis '16
Constraints	305	70	207
$\mathcal{I}^{Q}_{\mathsf{MI}}$ (or # of *mqis*)	28954	25303	7736

INCONSISTENCY MEASUREMENT

INDIVIDUAL FORMULAS

- Culpability Measure $C_{\#}$
- Let $\mathcal{B}$ be the set of all possible business rule bases, and $\mathcal{F}$ be the set of individual formulas that appear in a specific rule base in $\mathcal{B}$, then a culpability measure C is defined as

$$C: \mathcal{B} \times \mathcal{F} \longmapsto [0,\infty[$$



- $C_{\#}$ measure, defined via

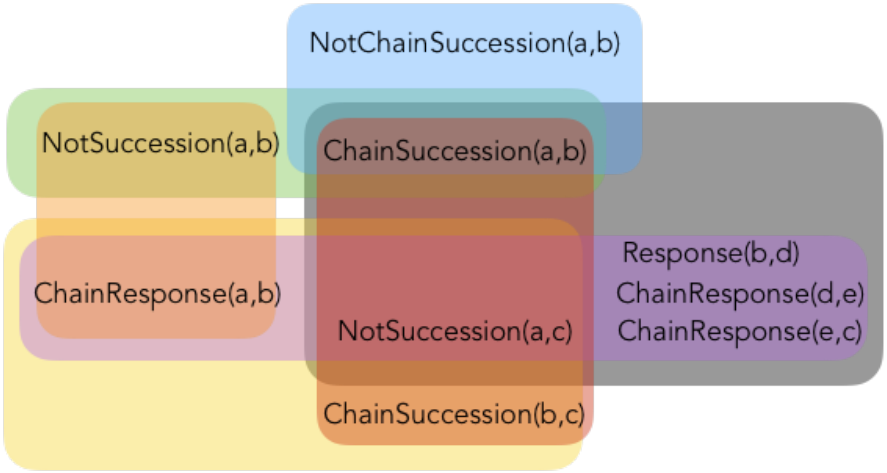
$$C_{\#}(RB,f) = |\{m \in MCS(RB) \mid f \in m\}|$$

- (i.e., the number of *MCSes* which *f* is part of)
- i.e., $C_{\#}(Example, ChainSuccession(a,b)) = 4$

INCONSISTENCY MEASUREMENT

INDIVIDUAL FORMULAS

- Values from culpability measures allow to quantify the severity of inconsistency for individual elements
  - C#(Example, ChainSuccession(a,b)) = 4
  - C#(Example, NotSuccession(a,c)) = 4
  - C#(Example, ChainResponse(a,b)) = 3
  - And so on
- Idea: Ranking



INCONSISTENCY MEASUREMENT

RANKING

- Definition (Culpability Ranking): Let *RB* be a rule base and *C* be a culpability measure. Then, a culpability ranking is any ranking over all $f_i \in RB$ via $\langle f_1, \ldots, f_n \rangle$, which satisfies $C(RB,f_1) \geq \ldots \geq C(RB,f_n)$

- Ranking can be used as a driver for re-modeling (Recommendations)
- Options for re-modeling
  - Discard rule
  - Change rule

COMPLIANCE MANAGEMENT STRATEGIES

- Design-Time Compliance
  - During modelling
  - No case data is known
- Run-Time Compliance
  - Case-dependent data has to be considered
- Post-Execution Compliance
  - → Lecture Smart Process Analytics

COMPLIANCE MANAGEMENT STRATEGIES

- Design-Time Compliance

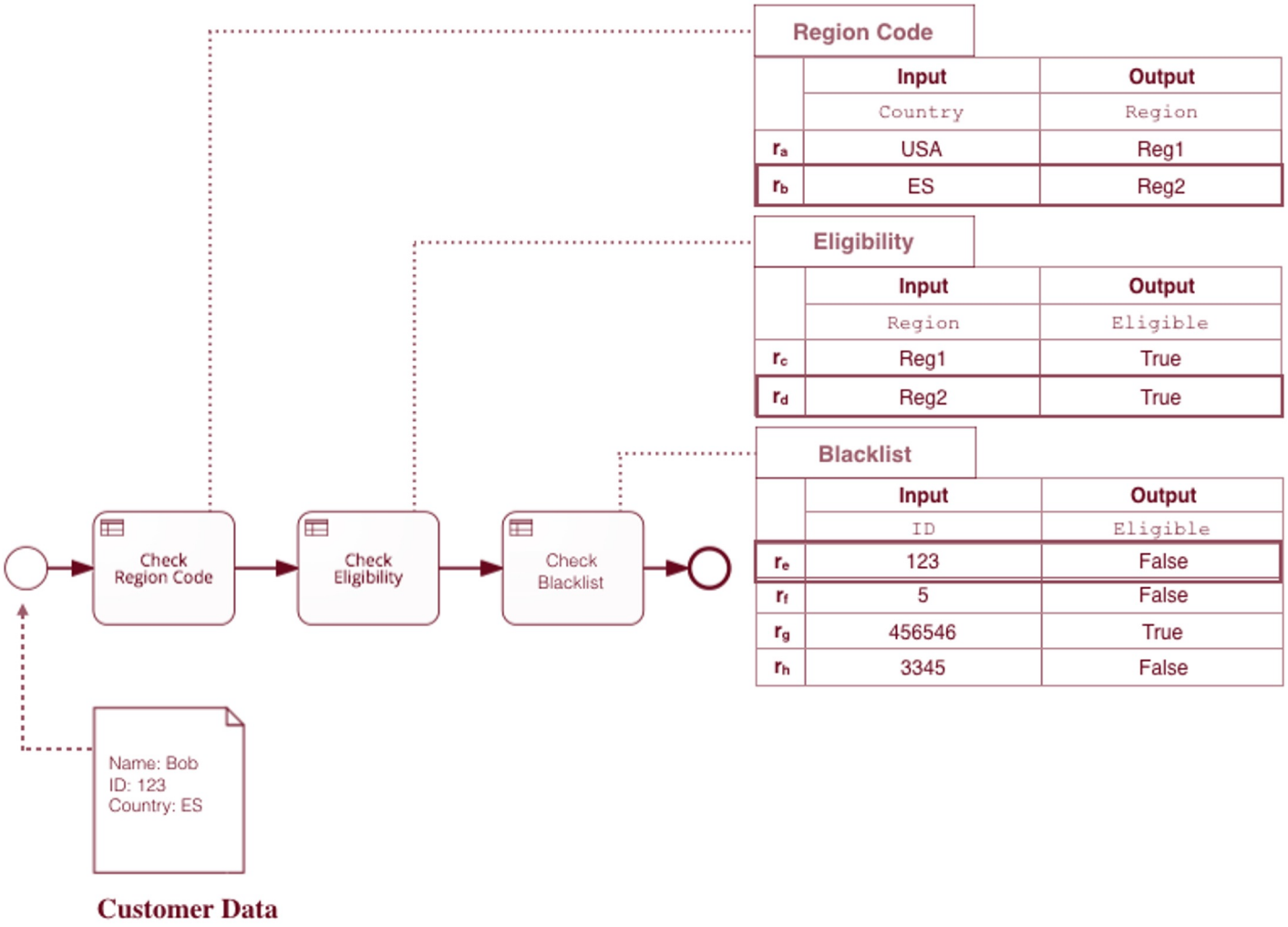
Subsumption 1

Decision_1vo386g

U	Input +	Output +
	-	-
	integer	string
1	[0..10]	"Hello"
2	[0..5]	"World"
3	[7..9]	"Hello"
4	[7..15]	"FuBa"

COMPLIANCE MANAGEMENT STRATEGIES

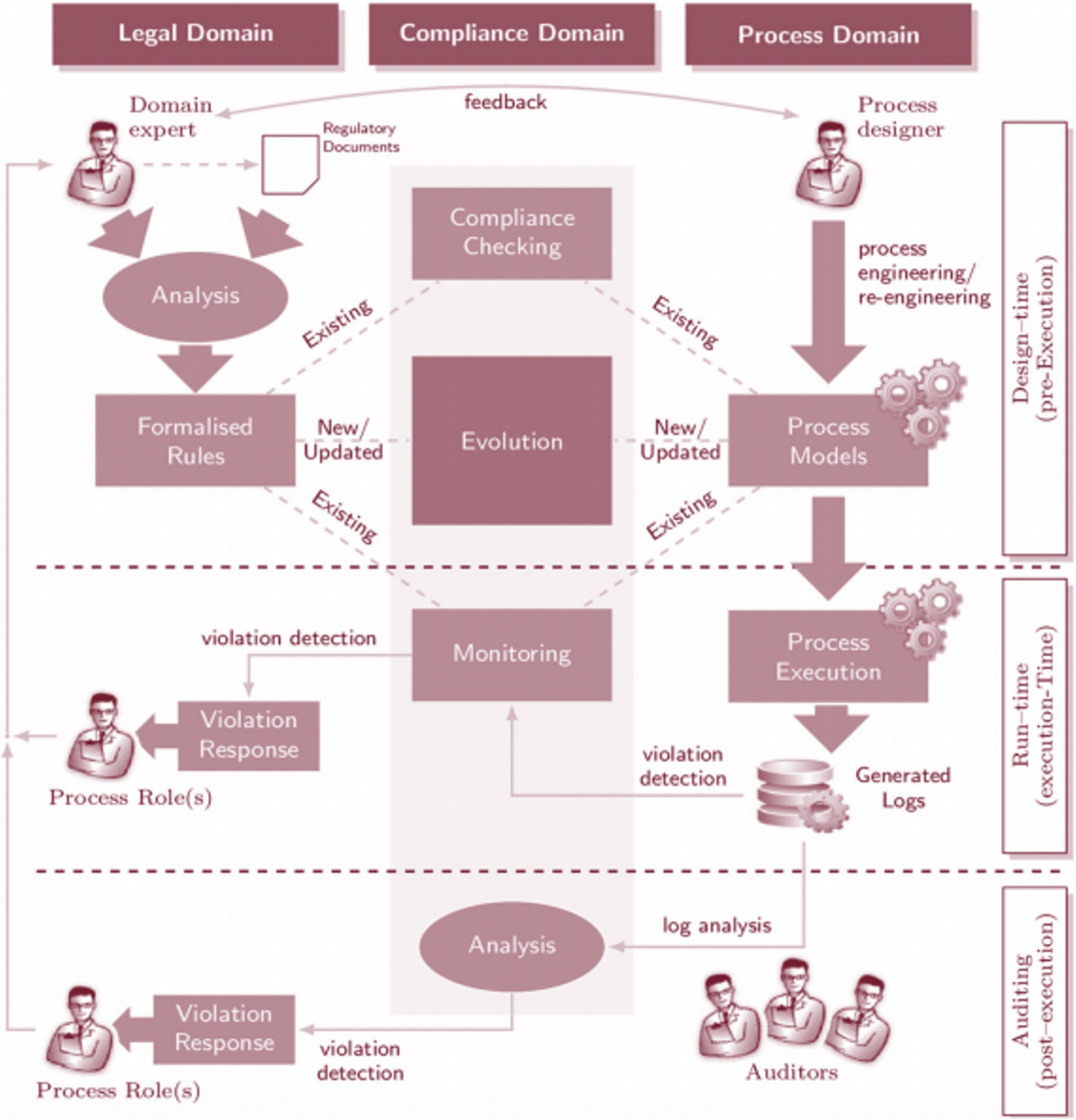
- Run-Time Compliance



DECISION LOGIC LEVEL VERIFICATION

DEMO

COMPLIANCE MANAGEMENT STRATEGIES

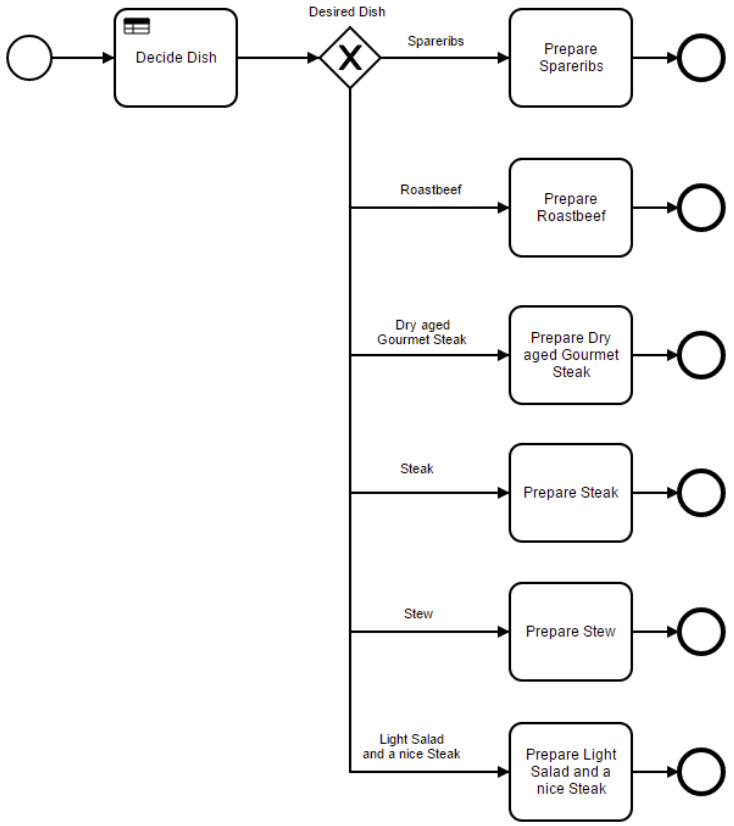


AGENDA

- Recap
- Business Rules Management
  - Organizing
    - .. In DMN
    - .. In Declare
  - Implementing

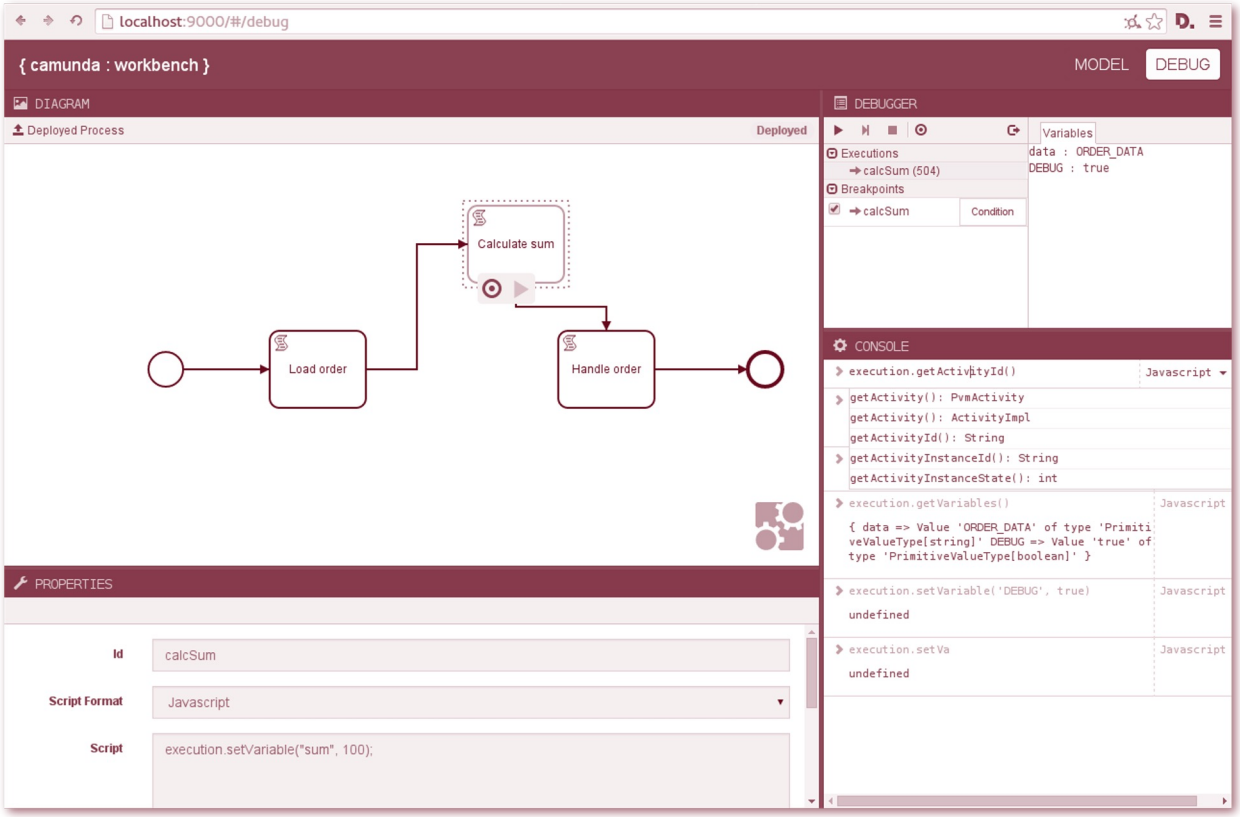
IMPLEMENTING

- Tasks in applying:
  - Integrating the formalized (and checked) rules in resp. Systems
    - E.g. Camunda



IMPLEMENTING

- Lecture → Workflow Management



IMPLEMENTING

- Tasks in applying:
  - Integrating the formalized (and checked) rules in resp. Systems
    - E.g. Camunda
  - Implementing BI and Monitoring

RECAP

- Organize:
  - Manage errors made in authoring
    - Detection
    - Quantify Errors (Inconsistency Measurement)
- Apply:
  - C.f. also the lecture Workflow Management

REFERENCES

- Hashmi, M., Governatori, G., Lam, H. P., & Wynn, M. T. (2018). Are we done with business process compliance: state of the art and challenges ahead. *Knowledge and Information Systems*, 1-55.
- Smit, K., Zoet, M., & Berkhout, M. (2017, July). Verification capabilities for business rules management in the Dutch governmental context. In *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)* (pp. 1-6). IEEE.
- Corea, C., Deisen, M., & Delfmann, P. Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement. In Proceedings der 14. Internationalen Tagung der Wirtschaftsinformatik (WI2019). Siegen, 2019
- Carl Corea, Patrick Delfmann. A Tool to Monitor Consistent Decision-Making in Business Process Execution. In Proceedings of the Demonstration Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM 2018). Sydney, 2018.

UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT

BUSINESS RULES & BUSINESS RULES MANAGEMENT

INSTITUTE FOR IS RESEARCH

www.uni-koblenz.de